

AD-A062810

LEVEL

R-1138

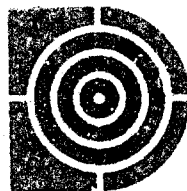
**A NEW BASELINE FOR THE INERTIAL NAVIGATION
STRAPDOWN SIMULATOR PROGRAM**

VOLUME IV Program Listings

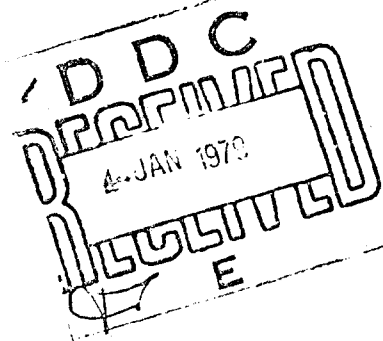
by

R.J. Neves, J.T. Prohaska, D.G. Riegler

July 1978



The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139



78 12 3 025

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DDC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM																				
1. REPORT NUMBER R-1136	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER																				
4. TITLE (and Subtitle) A New Baseline for the Inertial Navigation Strapdown Simulator Volumes I, II, III, and IV		5. TYPE OF REPORT & PERIOD COVERED 6/1/77 - 7/15/78																				
		6. PERFORMING ORG. REPORT NUMBER																				
7. AUTHOR(s) R. Nurse, J. Prohaska, D. Riegsecker		8. CONTRACT OR GRANT NUMBER(s) F33615-75-C-1149																				
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Charles Stark Draper Laboratory, Inc. 555 Technology Sq., Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project 6095, Task 4.2.5																				
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Avionics Laboratory Wright-Patterson AFB, Dayton, Ohio 45433		12. REPORT DATE July 1978																				
		13. NUMBER OF PAGES																				
14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED																				
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE																				
16. DISTRIBUTION STATEMENT (of this Report) APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED.		<table border="1"> <tr> <td colspan="2">ACCESSION for</td> </tr> <tr> <td>NTIS</td> <td>White Section <input checked="" type="checkbox"/></td> </tr> <tr> <td>DDC</td> <td>Buff Section <input type="checkbox"/></td> </tr> <tr> <td>UNANNOUNCED</td> <td><input type="checkbox"/></td> </tr> <tr> <td colspan="2">JUSTIFICATION</td> </tr> <tr> <td colspan="2">BY</td> </tr> <tr> <td colspan="2">DISTRIBUTION AVAILABILITY CODES</td> </tr> <tr> <td colspan="2">OR</td> </tr> <tr> <td colspan="2">SPECIAL</td> </tr> <tr> <td colspan="2">A 23 </td> </tr> </table>	ACCESSION for		NTIS	White Section <input checked="" type="checkbox"/>	DDC	Buff Section <input type="checkbox"/>	UNANNOUNCED	<input type="checkbox"/>	JUSTIFICATION		BY		DISTRIBUTION AVAILABILITY CODES		OR		SPECIAL		A 23	
ACCESSION for																						
NTIS	White Section <input checked="" type="checkbox"/>																					
DDC	Buff Section <input type="checkbox"/>																					
UNANNOUNCED	<input type="checkbox"/>																					
JUSTIFICATION																						
BY																						
DISTRIBUTION AVAILABILITY CODES																						
OR																						
SPECIAL																						
A 23																						
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)																						
18. SUPPLEMENTARY NOTES																						
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Inertial Navigation Laser Gyros Strapdown Random Vibration Simulation Single degree of freedom gyros and accelerometers Instrument Modeling																						
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This four volume report describes an updated and expanded version of a direct, digital, modular simulation of a strapdown inertial navigation system employing a wander-azimuth computational frame, and subject to a six degree of freedom random vibration environment. The original version of this simulation was developed under Task 4.2.3(a) of the above contract during 1975 and 1976. (CONTINUED ON REVERSE)																						

21-122376

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

The user may simulate not only the gross dynamics of the flight profile (from an external or internal profile generation) but also the angular and linear random vibrations resulting from gusts and turbulence acting on the airframe. The total environment is applied to the models of the inertial components (laser or SDR gyros and pendulous accelerometers). The resulting outputs of simulated IMU are summed in an interface module and compensated and scaled in the simulated navigation computer. The latter also contains the velocity/attitude algorithm, which computes the body-to-inertial transformation, using either the direction cosine matrix or quaternion, and the navigation algorithm which numerically integrates the specific forces after transformation to the local vertical, wander azimuth computational frame. The outputs of the simulated navigation computer are the computed position, velocity, and attitude of the vehicle with respect to a local vertical, north pointing frame. The flight profile and the differences between it and the simulated navigation computer outputs are tabulated in an evaluation module for printing, plotting, or post processing.

A ground alignment Kalman filter for the INSS, also developed under this task, is not documented in this report, but may be available from AFAL/RWA-2 or -3.

The program is written in Fortran IV for use on a CD6600/CYBER74.

The report is structured as follows:

- Volume I is the Introduction and Summary
- Volume II contains analytical development of the equations to be mechanized and the transition to difference equation form
- Volume III is the Program Description and User's Guide
- Volume IV contains Program Listings.

R-1136

A NEW BASELINE FOR THE INERTIAL NAVIGATION
STRAPDOWN SIMULATOR PROGRAM

VOLUME IV Program Listings

by

R.J. Nurse, J.T. Prohaska, D.G. Riegsecker

July 1978

Approved: W. Denhard
W. Denhard

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

ACKNOWLEDGEMENT

This four-volume report was prepared under USAF Contract No. F33615-75-C-1149 by the Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts, in accordance with Section 4 of the contract. The monitoring Air Force project engineer is Mr. Neil Banke (RWA-2), Air Force Avionics Laboratory, Dayton, Ohio.

The Draper Laboratory program manager for this task is John Harper and the lead engineer is Roy Nurse. The authors of the report are Roy Nurse, John Prohaska, and Darold Riegsecker.

The authors express their appreciation to Messrs. William Shephard, David Kaiser, Stanton Musick, and Maj. Robert Edwards of the Air Force Avionics Laboratory and William Ostanek, Jane Goode, Tom Thorvaldsen, and William Caffery of the C. S. Draper Laboratory for their assistance during the course of this contract.

VOLUME IV
TABLE OF CONTENTS

<u>SECTION</u>	<u>PAGE</u>
Sequencer Module (SEQ).....	1-2
Trajectory Module (TRAJ).....	1-11
Environment Module (ENV).....	1-19
Accelerometer Module (ACCEL).....	1-29
Gyro Module (GYROS).....	1-37
Laser Gyro Module (GYROS).....	1-44
Altimeter (ALTI).....	1-48
Hardware/Software Interface (RDR).....	1-51
Accelerometer Compensation (ACOMP).....	1-53
Gyro Compensation (GCOMP).....	1-58
Laser Gyro Compensation (GCOMP).....	1-64
Attitude and Velocity Algorithm (ALG).....	1-67
Local Level Navigator (LLN).....	1-76
Angular Velocity (ANGVEL).....	1-85
Angular Velocity by DT and Coriolis Correction (TORCOR).....	1-86
Second Order Update (AUP).....	1-87
Gravity Computation (GRAV).....	1-88
Matrix Multiply (MM).....	1-89
Evaluation (EVL).....	1-90
Mathematical Subroutines.....	1-94

PROGRAM LISTINGS

PROGRAM INSSEQ (INPUT,OUTPUT,TAPE0,TAPE6=OUTPUT,TAPE7,TAPE10,	00000010
1 TAPE12,TAPE20,TAPE30,TAPE40,TAPE50,TAPE60,TAPE65,TAPE67,TAPE69,	00000020
2 TAPE70,TAPE80,TAPE90,TAPE5)	00000030
C	00000040
C 03/10/78 DATE OF CURRENT MODULE	00000050
C	00000060
C THE SEQUENCER MODULE, THE EXECUTIVE, IS THE MAIN PROGRAM, WHICH CALLS	00000070
C THE OTHER MODULES	00000080
C	00000090
REAL ABB(3)	00000100
REAL AB(3)	00000110
REAL ALT	00000120
REAL ALTO	00000130
REAL BUF1(3)	00000140
REAL BUF2(3)	00000150
REAL BUF3(9)	00000160
REAL DATA(9)	00000170
REAL DTHETA(3)	00000180
REAL DTHETO(3)	00000190
REAL DTHETZ(3)	00000200
REAL DT	00000210
REAL DTSLOW	00000220
REAL DVI(3)	00000230
REAL DVA(3)	00000240
REAL DVNI(3)	00000250
REAL DVO(3)	00000260
REAL DVT(3)	00000270
REAL FT	00000280
REAL G	00000290
REAL HDING	00000300
REAL LAT	00000310
REAL LON	00000320
REAL HOOPDY	00000330
REAL NAVH	00000340
REAL NAVLAT	00000350
REAL NAVLON	00000360
REAL NAVV(3)	00000370
REAL OUTPSW	00000380
REAL PCUP(16)	00000390
REAL PDATA(20)	00000400
REAL PITCH	00000410
REAL PRNTDT	00000420
REAL PRNTSW	00000430
REAL RE	00000440
REAL ROLL	00000450
REAL SPARE1	00000460
REAL SPARE2	00000470
REAL SPARE3	00000480
REAL T	00000490
REAL TENO	00000500
REAL TPRMOD	00000510
REAL TPRNT	00000520
REAL TSEQ	00000530
REAL VEL(3)	00000540
REAL WBI(3)	00000550
REAL WBI(3)	00000560
REAL WBOOT(3)	00000570
REAL WE	00000580
REAL WONDER	00000590
REAL X1	00000600

	REAL X2	00000610
	REAL X3	00000620
	REAL X4	00000630
	REAL YAW	00000640
C		00000650
	REAL DCC(9)	00000660
	REAL DCM(9)	00000670
	REAL TESTID(9)	00000680
	REAL NAVP	00000690
	REAL NAVR	00000700
	REAL NAVHD	00000710
C		00000720
	INTEGER I	00000730
	INTEGER IENDF	00000740
	INTEGER IF1	00000750
	INTEGER IFILE	00000760
	INTEGER OFILE	00000770
	INTEGER PFILE	00000780
	INTEGER SIMEND	00000790
C		00000800
	EQUIVALENCE (DATA(1), DT)	00000810
	EQUIVALENCE (DATA(2), PRHTSW)	00000820
	EQUIVALENCE (DATA(3), OUTPSW)	00000830
	EQUIVALENCE (DATA(4), XFILE)	00000840
	EQUIVALENCE (DATA(5), SPARE1)	00000850
	EQUIVALENCE (DATA(6), SPARE2)	00000860
	EQUIVALENCE (DATA(7), TEND)	00000870
	EQUIVALENCE (DATA(8), DTSLOW)	00000880
	EQUIVALENCE (DATA(9), MODPDT)	00000890
C		00000900
	EQUIVALENCE (PDATA(1), HE)	00000910
	EQUIVALENCE (PDATA(2), RE)	00000920
	EQUIVALENCE (PDATA(3), G)	00000930
	EQUIVALENCE (PDATA(4), PRNTOT)	00000940
	EQUIVALENCE (PDATA(5), PDUP(1))	00000950
C		00000960
	DATA AOB/3*0.0/	00000970
	DATA AO/3*0.0/	00000980
	DATA ALT/0.0/	00000990
	DATA ALTO/0.0/	00001000
	DATA DATA/0*0.0/	00001010
	DATA DTHETA/3*0.0/	00001020
	DATA DTHETO/3*0.0/	00001030
	DATA DTHETZ/3*0.0/	00001040
	DATA DV/3*0.0/	00001050
	DATA DVA/3*0.0/	00001060
	DATA DVN/3*0.0/	00001070
	DATA DVO/3*0.0/	00001080
	DATA DVT/3*0.0/	00001090
	DATA FY/0.0/	00001100
	DATA LAT/0.0/	00001110
	DATA LON/0.0/	00001120
	DATA NAVH/0.0/	00001130
	DATA NAVLAT/0.0/	00001140
	DATA NAVLON/0.0/	00001150
	DATA NAVV/3*0.0/	00001160
	DATA PDATA/20*0.0/	00001170
	DATA PITCH/0.0/	00001180
	DATA ROLL/0.0/	00001190
	DATA T/0.0/	00001200

DATA TPRMOD/0.0/	00001210
DATA TPRNT/0.0/	00001220
DATA TSEQ/0.0/	00001230
DATA VEL/3*0.0/	00001240
DATA WB/3*0.0/	00001250
DATA WBB/3*0.0/	00001260
DATA WBBDOT/3*0.0/	00001270
DATA WORDER/0.0/	00001280
DATA NAVP/0.0/	00001290
DATA NAVR/0.0/	00001300
DATA NAVHD/0.0/	00001310
DATA IENDF/0/	00001320
DATA IFILE/10/	00001330
DATA PFILE/7/	00001340
DATA SIMENS/0/	00001350
C	00001360
C READ IN TEST IDENTIFICATION	00001370
C	00001380
REWIND 5	00001390
READ (5,507) TESTID	00001400
C	00001410
C***SEQUENCER INITIALIZATION***	00001420
C	00001430
REWIND IFILE	00001440
REWIND PFILE	00001450
1 CONTINUE	00001460
READ (IFILE,503) I,DATA(I)	00001470
IF (EOF(IFILE)) 2,1	00001480
2 CONTINUE	00001490
5 READ (PFILE,503) I,PODATA(I)	00001500
IF (EOF(PFILE)) 6,5	00001510
6 CONTINUE	00001520
REWIND IFILE	00001530
REWIND PFILE	00001540
C	00001550
C INITIALIZATION OUTPUT AND PRINT CONTROL	00001560
C	00001570
OFIL=XFILE	00001580
WRITE (OFIL,506) TESTID	00001590
WRITE (OFIL,505) W,PE,G,PRNTDT	00001600
WRITE (OFIL,500) DT, PRNTSW, OUTPSW, OFIL,	00001610
X PRNTDT,HOOPDT,TEHD,DYSLOW	00001620
C	00001630
C	00001640
T=0.	00001650
GO TO 4	00001660
C	00001670
C***EVERY CYCLE OF THE SEQUENCER BEGINS HERE***	00001680
C	00001690
3 CONTINUE	00001700
C	00001710
IF (PRNTDT.GT.0) GO TO 7	00001720
IF (HOOPDT.EQ.0) GO TO 4	00001730
C	00001740
IF (T.LT.TPRMOD-.0005) GO TO 4	00001750
TPRMOD=TPRMOD+HOOPDT	00001760
GO TO 8	00001770
C	00001780
7 CONTINUE	00001790
IF (T.LT.TPRNT-.0005) GO TO 4	00001800

C	TPRNT=TPRNT+PRNTDT	00001810
C	8 CONTINUE	00001820
	IF (PRNTSW.LT.1.) GO TO 4	00001830
	WRITE (OFIL,506) TESTID	00001840
	WRITE (OFIL,504) Y	00001850
C	4 CONTINUE	00001860
C		00001870
C		00001880
C		00001890
C		00001900
C		00001910
C	***TRAJECTORY MODULE***	00001920
C	INPUT: TIME	00001930
C	T	00001940
C		00001950
C	OUTPUT: LATITUDE, LONGITUDE, ALTITUDE, VELOCITY, DELTA VELOCITY,	00001960
C	PITCH, YAW, ROLL, HEADING, WANDER, BODY ACCEL, BODY RATES,	00001970
C	INERTIAL TO BODY MATRIX	00001980
C	LAT, LON, ALT, VEL, PITCH, YAW, ROLL, DVT, HDING, WANDER, AB, WB	00001990
C		00002000
	X1 = T	00002010
	IF1=IENDF	00002020
	BUF1(1)=PRNTSW	00002030
	BUF1(2)=MODPOT	00002040
C		00002050
	CALL TRAJ (X1, IENDF, PRNTSW, MODPOT,	00002060
	X LAT, LON, ALT, VEL, PITCH, YAW, ROLL, DVT, HDING, WANDER, AB, WB)	00002070
C		00002080
	IF (IENDF .NE. 0) SIMEND = 1	00002090
C		00002100
C		00002110
C		00002120
C		00002130
C	***ENVIRONMENT MODULE***	00002140
C		00002150
C	INPUT: TIME, GRAVITY WRT BODY, BODY RATES, INERTIAL TO BODY MAT	00002160
C	T, AB, WB	00002170
C		00002180
C	OUTPUT: INPUT VALUES WITH VIBRATION, BODY ACCELERATION, QBI + VIBRATION	00002190
C	ABB, WBB, WBBDOT	00002200
C		00002210
	X1 = T	00002220
	IF1 = IENDF	00002230
C		00002240
	DO 10 I=1,3	00002250
	BUF1(I)=AB(I)	00002260
	BUF2(I)=WB(I)	00002270
10	CONTINUE	00002280
C		00002290
C		00002300
	CALL ENV(X1, IF1, BUF1, BUF2,	00002310
	X ABB, WBB, WBBDOT)	00002320
C		00002330
	IF (IF1 .NE. 0) SIMEND=1	00002340
C		00002350
C		00002360
C		00002370
C	***GYRO MODULE***	00002380
C	OPERATES ON .001 SECOND CYCLE	00002390
C		00002400

C	INPUT: TIME,BODY RATES,CHANGE IN BODY RATES, BODY ACCELATION	00002410
C	T,WJB,WBBDOT,ARB	00002420
C		00002430
C	OUTPUT: GYRO PERTURBATIONS	00002440
C	DTHETA	00002450
C		00002460
	X1=T	00002470
	IF1=IENDF	00002480
	DO 20 I=1,3	00002490
	BUF1(I)=WBB(I)	00002500
	BUF2(I)=WBBDOT(I)	00002510
	BUF3(I)=ACB(I)	00002520
20	CONTINUE	00002530
C		00002540
	CALL GYROS (X1, IF1, BUF1, BUF2, BUF3,	00002550
X	DTHETA)	00002560
	IF (IF1 .NE. 0) SIMEND=1	00002570
C		00002580
C		00002590
C		00002600
C	***ACCELEROMETER MODULE***	00002610
C		00002620
C	INPUT: TIME,BODY ACCELATION,BODY RATES,CHANGE IN BODY RATE	00002630
C	T,ABB,WBB,WBBDOT	00002640
C		00002650
C	OUTPUT: DELTA VELOCITY	00002660
C	DV	00002670
C		00002680
	X1=T	00002690
	IF1=IENDF	00002700
	DO 30 I=1,3	00002710
	BUF1(I)=ABB(I)	00002720
	BUF2(I)=WBB(I)	00002730
	BUF3(I)=WBBDOT(I)	00002740
30	CONTINUE	00002750
C		00002760
	CALL ACCEL (X1,IF1,BUF1,BUF2,BUF3,	00002770
X	DV)	00002780
C		00002790
	IF (IF1 .NE. 0) SIMEND=1	00002800
C		00002810
C		00002820
C	END OF SEQUENCER FAST CYCLE	00002830
C		00002840
C	IF SIMULATION LAST PASS (IENDF=1) DONT CYCLE FAST CYCLE,	00002850
C	BUT FALL INTO SLOW CYCLE AND TERMINATE	00002860
C		00002870
	IF (IENDF.EQ.1) GO TO 39	00002880
C		00002890
	IF (T.LY.TSEQ-.0005) GO TO 72	00002900
	TSEQ=TSEQ+DTSLOW	00002910
C		00002920
C		00002930
	39 CONTINUE	00002940
C		00002950
C	THE FOLLOWING MODULES OPERATE ON THE SLOW CYCLE	00002960
C		00002970
C		00002980
C		00002990
C	***ALTIMETER MODULE***	00003000

C		00003010
C	INPUT: TIME, TRUE ALTITUDE, VELOCITY	00003020
C	T,ALT,VEL	00003030
C		00003040
C	OUTPUT: ALTITUDE	00003050
C	ALTO	00003060
C		00003070
C		00003080
	X1=T	00003090
	IF1=IENDF	00003100
	DO 40 I=1,3	00003110
	BUF1(I)=VEL(I)	00003120
40	CONTINUE	00003130
	BUF2(1)=ALT	00003140
C		00003150
	CALL ALTI (X1,IF1,BUF2(1),BUF1,	00003160
	X ALTO)	00003170
C		00003180
	IF (IF1 .NE. 0) SIMEND=1	00003190
C		00003200
C		00003210
C		00003220
C	***READER MODULE***	00003230
C		00003240
C	INPUT: TIME, GYRO PERTURBATIONS, DELTA VELOCITY	00003250
C	T,DTHETA,DV	00003260
C		00003270
C	THE READER INPUT ITEMS DTHETA AND DV ARE OUTPUT AS	00003280
C	ZERO QUANTITIES	00003290
C		00003300
C	OUTPUT: GIMBAL ANGLES, DELTA VELOCITY	00003310
C	DTHETO,DVO	00003320
C		00003330
C		00003340
C		00003350
	X1=T	00003360
	IF1=IENDF	00003370
C		00003380
	CALL RDR(X1,IF1,DTHETA,DV,	00003390
	X DTHETO,DVO)	00003400
C		00003410
	IF (IF1.NE.0) SIMEND=1	00003420
C		00003430
C		00003440
C		00003450
C	***ACCELEROMETER COMPENSATION MODULE***	00003460
C		00003470
C		00003480
C	INPUT: TIME, DELTA VELOCITY	00003490
C	T,DVO,DTHETO	00003500
C		00003510
C	OUTPUT: COMPENSATED DELTA VELOCITY	00003520
C	DVA	00003530
C		00003540
	X1=T	00003550
	IF1=IENDF	00003560
:	DO 45 I=1,3	00003570
	BUF1(I)=DVO(I)	00003580
	BUF2(I)=DTHETO(I)	00003590
45	CONTINUE	00003600

C		00003610
	CALL ACOMP(X1,IF1,BUF1,BUF2,	00003620
X	DVA)	00003630
C		00003640
	IF (IF1.NE.0) SIMEND=1	00003650
C		00003660
C		00003670
C		00003680
C	***GYRO COMPENSATION MODULE***	00003690
C		00003700
C	INPUT: TIME,GIMBAL ANGLES,COMPENSATED DELTA VELOCITY	00003710
C	T,DTHETO,DVO	00003720
C		00003730
C	OUTPUT: COMPENSATED GIMBAL ANGLES	00003740
C	DTHETZ	00003750
C		00003760
	X1=T	00003770
	IF1=IENDF	00003780
C		00003790
	DO 47 I=1,3	00003800
	BUF1(I)=DTHETO(I)	00003810
	BUF2(I)=DVA(I)	00003820
47	CONTINUE	00003830
C		00003840
	CALL GCOMP(X1,IF1,BUF1,BUF2,	00003850
X	DTHETZ)	00003860
C		00003870
	IF (IF1.NE.0) SIMEND=1	00003880
C		00003890
C		00003900
C		00003910
C	***ALGORITHM MODULE***	00003920
C		00003930
C	INPUT: TIME,COMPENSATED GIMBAL ANGLES, COMP. DEL VELOCITY	00003940
C	T,DTHETZ,DVA	00003950
C		00003960
C	OUTPUT: DELTA VELOCITY, DIRECTION COSINE MATRIX	00003970
C	DVN,DCM	00003980
C		00003990
	X1=T	00004000
	IF1=IENDF	00004010
	DO 50 I=1,3	00004020
	BUF1(I)=DTHETZ(I)	00004030
	BUF2(I)=DVA(I)	00004040
50	CONTINUE	00004050
C		00004060
	CALL ALG (X1,IF1,BUF1,BUF2,	00004070
X	DVN,DCM)	00004080
C		00004090
	IF (IF1 .NE. 0) SIMEND=1	00004100
C		00004110
C		00004120
C		00004130
C	***NAVIGATION MODULE***	00004140
C		00004150
C	INPUT: TIME, DELTA VELOCITY, ALTITUDE,DIRECTION COS MATRIX	00004160
C	T,DVN,ALTO,DCM	00004170
C		00004180
C	OUTPUT: LATITUDE, LONGITUDE, VELOCITY, ALTITUDE, LV RATES, ATTITUDE	00004190
C	NAVLAT, NAVLON, NAVV, NAVH, NAVP, NAVR, NAVHD	00004200

C		00004210
	X1=T	00004220
	IF1=IENDF	00004230
	X2=ALTO	00004240
C		00004250
	CALL LLN (X1,IF1,DVN,X2,DCM,	00004260
X	NAVLAT,NAVLON,NAVX,NAVH,NAVP,NAVR,NAVHD)	00004270
C		00004280
	IF (IF1.NE. 0) SIMEND=1	00004290
C		00004300
C		00004310
C		00004320
C	***EVALUATION MODULE***	00004330
C	COMPUTES AND PRINTS TABLE DATA EVERY 50 CYCLES	00004340
C		00004350
C	INPUT: TIME,TRAJECTORY,NAVIGATION,AND ATTITUDE PARAMETERS	00004360
C	T,LAT,LON,ALT,VEL,DVT,PITCH,ROLL,YAW,WONDER,NAVLAT,NAVLON,	00004370
C	NAVX,NAVH,NAVP,NAVR,NAVHD	00004380
C		00004390
C	OUTPUT: NOTHING	00004400
C		00004410
	X1=T	00004420
	IF1=IENDF	00004430
C		00004440
	CALL EVL (X1,IF1,LAT,LON,ALT,VEL,DVT,PITCH,ROLL,YAW,WONDER,	00004450
	XNAVLAT,NAVLON,NAVX,NAVH,NAVP,NAVR,NAVHD)	00004460
C		00004470
	IF (IF1.NE.0) SIMEND=1	00004480
C		00004490
C		00004500
72	CONTINUE	00004510
C		00004520
	IF (IENDF.EQ. 1) GO TO 1000	00004530
	IF (T.GE. TEND-.0005) GO TO 81	00004540
	IF (SIMEND.NE. 0) GO TO 81	00004550
	IF (T.EQ. 0) WRITE(OFIL,501)	00004560
C		00004570
	T=T+DT	00004580
	IT=T	00004590
	FT=T-IT	00004600
	IF (FT.LT. .999) GO TO 80	00004610
	T=IT+1.0	00004620
80	CONTINUE	00004630
C		00004640
	GO TO 3	00004650
C		00004660
81	CONTINUE	00004670
C		00004680
C	SIMULATION END TIME REACHED	00004690
C		00004700
	IENDF=1	00004710
C		00004720
	GO TO 3	00004730
C		00004740
1000	CONTINUE	00004750
	PRINT 502, T	00004760
:	STOP	00004770
C		00004780
500	FORMAT (30H SEQUENCER INITIALIZATION ,	00004790
X	/3X,8H DT ,3X,616.8,3X,4H SEC,	00004800

X	/3X,8H PRNTSW ,3X,G16.8,	00004810
X	/3X,8H OUTPSW ,3X,G16.8,	00004820
X	/3X,8H OFILE ,3X,I15,	00004830
X	/3X,8H PRNTDT ,3X,G16.8,	00004840
X	/3X,8H MODPDT ,3X,G16.8,	00004850
X	/3X,8H TEND ,3X,G16.8,	00004860
X	/3X,8H DTSLOW ,3X,G16.8,/))	00004870
501	FORMAT (1H1,25H ***START SIMULATION*** ,//)	00004880
502	FORMAT (//26H ***SIMULATION COMPLETE AT ,G16.8, 4H SEC)	00004890
503	FORMAT (15,F20.10)	00004900
504	FORMAT (5X,18H ** SEQ ** TIME= ,G16.8,//)	00004910
505	FORMAT (30H PHYSICAL DATA FILE ,	00004920
	X/3X,8H WE ,3X,G16.8,3X,8H RAD/SEC,	00004930
	X/3X,8H RE ,3X,G16.8,3X,8H FT ,	00004940
	X/3X,8H G ,3X,G16.8,3X,8H FT/SEC2,	00004950
	X/3X,8H PRNTDT ,3X,G16.8,3X,8H SECS ,///)	00004960
506	FORMAT (1H1,9A8,/))	00004970
507	FORMAT (9A8)	00004980
C		00004990
600	FORMAT (12X,I15)	00005000
	END	00005010

C		00000010
C	03/10/78 DATE OF CURRENT MODULE	00000020
C		00000030
C	THE INTERIM TRAJECTORY MODULE INTEGRATES THE STANDARD LOCAL	00000040
C	LEVEL NORTH-SLAVED NAVIGATION EQUATIONS USING THE WGS72 ELLIPSOID	00000050
C	EARTH MODEL AND FIRST ORDER ALGORITHMS THROUGHOUT, AS SUCH IT	00000060
C	CONTAINS PHASING ERRORS WHICH WILL CAUSE HIGHER ORDER ALGORITHMS	00000070
C	TO HAVE APPARENT ERRORS. NOTE THAT INITIAL VELOCITIES ARE INPUT	00000080
C	IN AN ENU-NORTHSLAVED FRAME.	00000090
C		00000100
	SUBROUTINE TRAJ (T,IENDF,TBUF1,TBUF2,	00000110
X	LAT,LON,ALT,VEL,PITCH,YAW,ROLL,DV,HDING,WANDER,AB,WB)	00000120
C		00000130
	REAL AB(3)	00000140
	REAL ADI(3)	00000150
	REAL ABP(3)	00000160
	REAL ALT	00000170
	REAL TBUF1	00000180
	REAL TBUF2	00000190
	REAL COSLN	00000200
	REAL COSLT	00000210
	REAL DATA(15)	00000220
	REAL DELON	00000230
	REAL DT	00000240
	REAL DV(3)	00000250
	REAL ESQ	00000260
	REAL FORS(3)	00000270
	REAL GR	00000280
	REAL GN	00000290
	REAL G	00000300
	REAL GVEC(3)	00000310
	REAL HDING	00000320
	REAL IALT	00000330
	REAL ILAT	00000340
	REAL ILON	00000350
	REAL INTLON	00000360
	REAL IPITCH	00000370
	REAL IROLL	00000380
	REAL ITEMP(9)	00000390
	REAL IVEL(3)	00000400
	REAL IYAW	00000410
	REAL LAT	00000420
	REAL LON	00000430
	REAL MODPDT	00000440
	REAL MTEMP(9)	00000450
	REAL PBUF(16)	00000460
	REAL PDATA(20)	00000470
	REAL PDOT	00000480
	REAL PITCH	00000490
	REAL PRNTDT	00000500
	REAL QBI(9)	00000510
	REAL QIBOLD(9)	00000520
	REAL QBP(9)	00000530
	REAL QEI(9)	00000540
	REAL QPE(9)	00000550
	REAL QCP(9)	00000560
	REAL QTEMP(9)	00000570
	REAL RM	00000580
	REAL RP	00000590
	REAL ROOT	00000600

REAL	RTODG	00000610
REAL	ROLL	00000620
REAL	RE	00000630
REAL	SINLN	00000640
REAL	SINLT	00000650
REAL	SPARE1	00000660
REAL	T	00000670
REAL	TPRMOD	00000680
REAL	TPRNT	00000690
REAL	TTRJ	00000700
REAL	VELOLD(3)	00000710
REAL	VEL(3)	00000720
REAL	W(3)	00000730
REAL	WE	00000740
REAL	WB(3)	00000750
REAL	WBP(3)	00000760
REAL	WSDOT(3)	00000770
REAL	WBOLD(3)	00000780
REAL	WANDER	00000790
REAL	WTEMP(9)	00000800
REAL	YAW	00000810
REAL	YDOT	00000820
REAL	IWANDR	00000830
REAL	CHET	00000840
REAL	SWET	00000850
REAL	CP	00000860
REAL	SP	00000870
REAL	CR	00000880
REAL	SR	00000890
REAL	CY	00000900
REAL	SY	00000910
REAL	SPGR	00000920
REAL	SPCR	00000930
REAL	SALF	00000940
REAL	CALF	00000950
REAL	QDC(9)	00000960
C		00000970
	INTEGER IENDF	00000980
	INTEGER INITSW	00000990
	INTEGER OUTSW	00001000
	INTEGER XFILE	00001010
	INTEGER PFILE	00001020
	INTEGER OFILE	00001030
C		00001040
	EQUIVALENCE (DATA(1), DT)	00001050
	EQUIVALENCE (DATA(2), PRNTSW)	00001060
	EQUIVALENCE (DATA(3), OUTSW)	00001070
	EQUIVALENCE (DATA(4), XFILE)	00001080
	EQUIVALENCE (DATA(5), ILAT)	00001090
	EQUIVALENCE (DATA(6), ILOH)	00001100
	EQUIVALENCE (DATA(7), IALT)	00001110
	EQUIVALENCE (DATA(8), IVEL(1))	00001120
	EQUIVALENCE (DATA(9), IVEL(2))	00001130
	EQUIVALENCE (DATA(10), IVEL(3))	00001140
	EQUIVALENCE (DATA(11), IPIYCH)	00001150
	EQUIVALENCE (DATA(12), IYAW)	00001160
	EQUIVALENCE (DATA(13), IROLL)	00001170
	EQUIVALENCE (DATA(14), MODPDT)	00001180
	EQUIVALENCE (DATA(15), IWANDR)	00001190
C		00001200

EQUIVALENCE (PDATA(1), WE)	00001210
EQUIVALENCE (PDATA(2), RE)	00001220
EQUIVALENCE (PDATA(3), G)	00001230
EQUIVALENCE (PDATA(4), PRNTDT)	00001240
EQUIVALENCE (PDATA(5), PBUF(1))	00001250
C	00001260
DATA ESQ/.006694317778/	00001270
DATA ROTODG/57.29577951/	00001280
DATA TFRMOD/0.0/	00001290
DATA TPRNT/0.0/	00001300
DATA INITSW/0/	00001310
DATA IFILE/20/	00001320
DATA PFILE/7/	00001330
C	00001340
IF (IENDF.EQ.1) RETURN	00001350
IF (INITSW.EQ.0) GO TO 500	00001360
IF (T.LT.TTRJ-.0005) RETURN	00001370
C	00001380
DO 110 I=1,3	00001390
VELOLD(I)=VEL(I)	00001400
110 CONTINUE	00001410
C	00001420
C	00001430
ALT=ALT+VEL(3)*DT	00001440
C	00001450
GR=-(32.0877057+.16939081*SINLT*SINLT+7.52810E-4*	00001460
X SINLT**4)*(1.-(9.6227E-8 -6.4089E-10*SINLT*SINLT)*	00001470
X ALT+6.8512E-15*ALT*ALT)	00001480
GN=1.63E-8*ALT*SINLT*COSLT	00001490
C	00001500
DV(3)=-((VEL(1)*(W(2)+2.*WE*COSLT)	00001510
X -VEL(2)*W(1)+GR)*DT+VEL(3)-VELOLD(3)	00001520
C	00001530
DV(2)=-((VEL(3)*W(1)-VEL(1)*(W(3)	00001540
X +2.*WE*SINLT)+GN)*DT+VEL(2)-VELOLD(2)	00001550
C	00001560
DV(1)=-((VEL(2)*(W(3)+2.*WE*SINLT))	00001570
X -VEL(3)*(W(2)+2.*WE*COSLT))*DT+VEL(1)-VELOLD(1)	00001580
C	00001590
C COMPUTE ACTING FORCES IN NAVIGATION COORDINATES (LOCAL VERTICAL)	00001600
C	00001610
DO 120 I=1,3	00001620
FORS(I)=DV(I)/DT	00001630
120 CONTINUE	00001640
C	00001650
RM=RE*(1.-ESQ)/((1.-ESQ*SINLT*SINLT)**1.5) + ALT	00001660
RP=RE/SQRT((1.-ESQ*SINLT*SINLT)) + ALT	00001670
C	00001680
C	00001690
W(1)=-VEL(2)/RM	00001700
W(2)=VEL(1)/RP	00001710
LAT=LAT-W(1)*DT	00001720
SINLT=SIN(LAT)	00001730
COSLT=COS(LAT)	00001740
W(3)=W(2)*SINLT/COSLT	00001750
WANDER = WANDER - W(3)*DT	00001760
SALF=SIN(WANDER)	00001770
CALF=COS(WANDER)	00001780
LON=LON+W(2)*DT/COSLT	00001790
SINLN=SIN(LON)	00001800

C	COSLN=COS(LON)	00001810
C	YDOT=-W(3)	00001820
C	YAW = YAW + YDOT*DT	00001830
C	STORE TRANSFORM OF QBI COMPUTED AT T-DT	00001840
C	QIBOLD(1)=QBI(1)	00001850
	QIBOLD(2)=QBI(4)	00001860
	QIBOLD(3)=QBI(7)	00001870
	QIBOLD(4)=QBI(2)	00001880
	QIBOLD(5)=QBI(5)	00001890
	QIBOLD(6)=QBI(8)	00001900
	QIBOLD(7)=QBI(3)	00001910
	QIBOLD(8)=QBI(6)	00001920
	QIBOLD(9)=QBI(9)	00001930
C	STORE BODY RATES COMPUTED AT T-DT	00001940
C	DO 130 I=1,3	00001950
	WBOLD(I)=WB(I)	00001960
	130 CONTINUE	00001970
C	GO COMPUTE INERTIAL TO BODY TRANSFORMATION (QBI)	00001980
C	GO TO 800	00001990
	200 CONTINUE	00002000
C	COMPUTE BODY RATES AND ACCELERATIONS IN BODY COORDINATES	00002010
C	CALL MXM(QBI,QIBOLD,MTEMP)	00002020
C	WB(1)=MTEMP(6)/DT	00002030
	WB(2)=-MTEMP(3)/DT	00002040
	WB(3)=MTEMP(2)/DT	00002050
C	CALL MXV(QBP,FORS,AB)	00002060
C	DO 220 I=1,3	00002070
	WBODT(I)=(WB(I)-WBOLD(I))/DT	00002080
	220 CONTINUE	00002090
C	ITEMP(1)=LAT*RDODG	00002100
	ITEMP(2)=LON*RDODG	00002110
	ITEMP(3)=WANDER*RDODG	00002120
C	OUTPUT AND PRINT CONTROL	00002130
C	IF (PRNTDT.GT.0) GO TO 960	00002140
	IF (MODPDT.EQ.0) GO TO 999	00002150
C	IF (T.LT.TPRMOD-.0005) GO TO 999	00002160
	TPRNT=TPRMOD+MODPDT	00002170
	GO TO 970	00002180
C	960 CONTINUE	00002190
	IF (T.LT.TPRNT-.0005) GO TO 999	00002200
	TPRNT=TPRNT+PRNTDT	00002210
		00002220
		00002230
		00002240
		00002250
		00002260
		00002270
		00002280
		00002290
		00002300
		00002310
		00002320
		00002330
		00002340
		00002350
		00002360
		00002370
		00002380
		00002390
		00002400

C		00002410
C		00002420
	970 CONTINUE	00002430
	IF (PRNTSW.LT.1.) GO TO 999	00002440
	WRITE(OFIL,1200) AB,WB,ITEMP(1),ITEMP(2),ITEMP(3),ALT,VEL,DV	00002450
C		00002460
	WRITE(OFIL,1300)QBX	00002470
C		00002480
C		00002490
	999 CONTINUE	00002500
	TTRJ=T*DT	00002510
	RETURN	00002520
C		00002530
C	INITIALIZATION BEGINS HERE	00002540
C		00002550
	500 CONTINUE	00002560
	REWIND IFIL	00002570
	REWIND PFILE	00002580
	501 READ (IFIL,1000) IX,DATA(IX)	00002590
	IF (EOF(IFIL)) 502,501	00002600
	502 CONTINUE	00002610
	503 READ (PFILE,1000) IX,PDATA(IX)	00002620
	IF (EOF(PFILE)) 510,503	00002630
	510 CONTINUE	00002640
	REWIND IFIL	00002650
	REWIND PFILE	00002660
C		00002670
	OFIL=XFIL	00002680
C		00002690
C		00002700
	LAT=ILAT/RTODG	00002710
	LON=ILON/RTODG	00002720
	INTLON=LON	00002730
	SINLT=SIN(LAT)	00002740
	COSLT=COS(LAT)	00002750
	SINLN=SIN(LON)	00002760
	COSLN=COS(LON)	00002770
	ALT=IALT	00002780
	RH=RE*(1.-ESQ)/((1.-ESQ*SINLT*SINLT)**1.5) + ALT	00002790
	RP=RE/SQRT((1.-ESQ*SINLT*SINLT)) + ALT	00002800
	NDING=(IYAW-IWANDR)/RTODG	00002810
	WANDER=IWANDR/RTODG	00002820
	SALF=SIN(WANDER)	00002830
	CALF=COS(WANDER)	00002840
C		00002850
C		00002860
	DO 512 I=1,3	00002870
	VEL(I)=I*VEL(I)	00002880
	512 CONTINUE	00002890
C		00002900
	PITCH=IPITCH/RTODG	00002910
	YAW=IYAW/RTODG	00002920
	ROLL=IROLL/RTODG	00002930
C		00002940
	PDOY=0.0	00002950
	YDOY=0.0	00002960
	RDOY=0.0	00002970
C		00002980
	W(1)=-VEL(2)/RH	00002990
	W(2)=VEL(1)/RP	00003000

	W(3)=W(2)*SINLT/COSLT	00003010
C		00003020
C	COMPUTE INITIAL FORCE AB IN ENU FRAME	00003030
C		00003040
	GR=-(32.0877057+.16939081*SINLT*SINLT+7.52810E-4*	00003050
	X SINLT**4)*(1.-(9.6227E-8 -6.4089E-10*SINLT*SINLT)*	00003060
	X ALT+6.8512E-15*ALT*ALT)	00003070
	GN=1.63E-8*ALT*SINLT*COSLT	00003080
	ABP(1)=0.0	00003090
	ABP(2)=GN	00003100
	ABP(3)=GR	00003110
C		00003120
C	GO COMPUTE INITIAL INERTIAL TO BODY MATRIX (QBI)	00003130
C		00003140
	GO TO 800	00003150
600	CONTINUE	00003160
	CALL MXV(QBP,ABP,AB)	00003170
C		00003180
C	COMPUTE INITIAL BODY RATE WRT INERTIAL AND TRANSFORM TO BODY FRAME	00003190
C		00003200
	WBP(1)=-COSLN*VEL(2)/RM	00003210
	WBP(2)=VEL(1)/(COSLT*RP) + WE	00003220
	WBP(3)=SINLN*VEL(2)/RM	00003230
	CALL MXV(QBI,WBP,WB)	00003240
C		00003250
C	ADD INITIALIZATION TO PFILE	00003260
C		00003270
	PDATA(1)=WE	00003280
	PDATA(2)=RE	00003290
	PDATA(3)=G	00003300
	PDATA(4)=PRNTDT	00003310
	PDATA(5)=LAT	00003320
	PDATA(6)=LON	00003330
	PDATA(7)=WANDER	00003340
	PDATA(8)=ALT	00003350
	PDATA(9)=ROLL	00003360
	PDATA(10)=PITCH	00003370
	PDATA(11)=YAW	00003380
	PDATA(12)=RDOT	00003390
	PDATA(13)=PDOT	00003400
	PDATA(14)=YDOT	00003410
	PDATA(15)=VEL(1)	00003420
	PDATA(16)=VEL(2)	00003430
	PDATA(17)=VEL(3)	00003440
	PDATA(18)=AB(1)	00003450
	PDATA(19)=AB(2)	00003460
	PDATA(20)=AB(3)	00003470
C		00003480
	WRITE(PFILE,1000) (I,PDATA(I),I=1,20)	00003490
C		00003500
C		00003510
C	INITIALIZATION OUTPUT AND PRINT CONTROL	00003520
C		00003530
	WRITE(OFIL,1010)DT,PRNTSW,OUTSW,OFIL,MODPDY,PRNTDT,	00003540
	X ILAT,ILON,IALY,IVEL,IROLL,IPITCH,IYAW,IMANDR,AB,WB	00003550
C		00003560
C		00003570
	INITSW=1	00003580
	TTRJ=T+DT	00003590
	RETURN	00003600

C		00003610
C		00003620
C	800 CONTINUE	00003630
C		00003640
C	COMPUTE EARTH TO PLATFORM TRANSFORMATION MATRIX (QPE)	00003650
C		00003660
	QPE(1)=COSLN	00003670
	QPE(2)=0.0	00003680
	QPE(3)=-SINLN	00003690
	QPE(4)=-SINLT*SINLN	00003700
	QPE(5)=COSLT	00003710
	QPE(6)=-SINLT*COSLN	00003720
	QPE(7)=COSLT*SINLN	00003730
	QPE(8)=SINLT	00003740
	QPE(9)=COSLT*COSLN	00003750
C	WRITE (OFILE,1300) QPE	00003760
C		00003770
C	COMPUTE PLATFORM TO BODY TRANSFORMATION MATRIX (QBP)	00003780
C	ONLY COMPUTED DURING INITIALIZATION OR WHEN YDOT IS NOT ZERO	00003790
C		00003800
	IF (INITSW .EQ. 1) GO TO 805	00003810
	QEI(2)=0.0	00003820
	QEI(4)=0.0	00003830
	QEI(5)=1.0	00003840
	QEI(6)=0.0	00003850
	QEI(8)=0.0	00003860
	QCP(3)=0.0	00003870
	QCP(6)=0.0	00003880
	QCP(7)=0.0	00003890
	QCP(8)=0.0	00003900
	QCP(9)=1.0	00003910
C		00003920
	SP=SIN(PITCH)	00003930
	CP=COS(PITCH)	00003940
	SR=SIN(ROLL)	00003950
	CR=COS(ROLL)	00003960
	SPSR=SP*SR	00003970
	SPCR=SP*CR	00003980
	GO TO 810	00003990
C		00004000
	805 CONTINUE	00004010
	IF (YDOT .EQ. 0.0) GO TO 815	00004020
C		00004030
	810 CONTINUE	00004040
	SY=SIN(YAW)	00004050
	CY=COS(YAW)	00004060
C		00004070
	QBC(1)=SY*CP	00004080
	QBC(2)=CY*CP	00004090
	QBC(3)=SP	00004100
	QBC(4)=-SY*SPSR-CY*CR	00004110
	QBC(5)=-CY*SPSR+SY*CR	00004120
	QBC(6)=CP*SR	00004130
	QBC(7)=-SY*SPCR+CY*SR	00004140
	QBC(8)=-CY*SPCR-SY*SR	00004150
	QBC(9)=CP*CR	00004160
C	WRITE (OFILE,1300) QBC	00004170
C		00004180
C		00004190
C	COMPUTE INERTIAL TO EARTH TRANSFORMATION MATRIX (QEI)	00004200

C		00004210
	815 CONTINUE	00004220
	SWET=SIN(WE* π)	00004230
	CWET=cos(WE* π)	00004240
	QEI(1)=CWET	00004250
	QEI(3)=-SWET	00004260
	QEI(7)=SWET	00004270
	QEI(9)=CWET	00004280
C	WRITE (OFILE,1300) QEI	00004290
C		00004300
C	COMPUTE NORTH POINTING TO WANDER PLATFORM MATRIX (QCP)	00004310
C		00004320
	QCP(1)=CALF	00004330
	QCP(2)=SALF	00004340
	QCP(4)=-SALF	00004350
	QCP(5)=CALF	00004360
C	WRITE (OFILE,1300) QCP	00004370
C		00004380
C	COMPUTE INERTIAL TO BODY TRANSFORMATION MATRIX (QBI)	00004390
C		00004400
	CALL MXM(QSC,QCP,QBP)	00004410
C	WRITE (OFILE,1300) QBP	00004420
	CALL MXM(QBP,QPE,QTEMP)	00004430
C	WRITE (OFILE,1300) QTEMP	00004440
	CALL MXM(QTEMP,QEI,QBI)	00004450
C	WRITE (OFILE,1300) QBI	00004460
C		00004470
	IF (INITSW.EQ.1) GO TO 200	00004480
	GO TO 600	00004490
C		00004500
C		00004510
	1000 FORMAT (15,F20.10)	00004520
	1010 FORMAT (30N TRAJECTORY INITIALIZATION ,	00004530
	X/3X,0N DT ,3X,G16.8,3X,4N SEC,	00004540
	X/3X,0N PNTSW ,3X,G16.8,	00004550
	X/3X,0N OUTSW ,3X,G16.8,	00004560
	X/3X,0N OFILE ,3X,I15,	00004570
	X/3X,0N MODPDT ,3X,G16.8,	00004580
	X/3X,0N PRNTDT ,3X,G16.8,/,	00004590
	X/3X,15H LAT(DEG) ,3X,G16.8,	00004600
	X/3X,15H LON(DEG) ,3X,G16.8,	00004610
	X/3X,15H ALT(FT) ,3X,G16.8,	00004620
	X/3X,15H VEL(FT/SEC) ,3X,3G16.8,/,	00004630
	X/3X,15H ROLL(DEGS) ,G16.8,	00004640
	X/3X,15H PITCH(DEGS) ,G16.8,	00004650
	X/3X,15H YAW(DEGS) ,G16.8,	00004660
	X/3X,15H WANDER(DEGS) ,G16.8,	00004670
	X/3X,15H AB(FT/SEC2) ,3G16.8,	00004680
	X/3X,15H W3(RADS/SEC) ,3G16.8,/,	00004690
	1100 FORMAT(2X,G16.8)	00004700
	1200 FORMAT(6X,25H ** TRJ ** AB(FT/SEC2) ,3G16.8,/ X 6X,25H WB(RAD/SEC) ,3G16.8,/ X 6X,25H LAT(DEGS) ,G16.8,/ X 6X,25H LON(DEGS) ,G16.8,/ X 6X,25H ALT(FT/SEC) ,G16.8,/ X 6X,25H VEL(FT/SEC) ,3G16.8,/ X 6X,25H ROLL(FT/SEC) ,3G16.8,/ 1300 FORMAT(12X,6H QBI ,3G16.8,/24X,3G16.8,/24X,3G16.8,/, 1310 FORMAT(10X,6H QBI ,3G16.8,/24X,3G16.8,/24X,3G16.8,/, END	00004710 00004720 00004730 00004740 00004750 00004760 00004770 00004780 00004790 00004800

C		00000010
C	05/03/78 DATE OF CURRENT MODULE	00000020
C		00000030
C	THE ENVIRONMENT MODULE IMPOSES VIBRATIONAL DISPLACEMENTS UPON THE	00000040
C	POSITION, ATTITUDE, AND VELOCITY COMPONENTS OF THE TRAJECTORY	00000050
C		00000060
	SUBROUTINE ENV (T,IENDF,AB,WB,	00000070
	X ABB,WBS,WBBDOT)	00000080
	REAL A	00000090
	REAL AB(3)	00000100
	REAL ABB(3)	00000110
	REAL AP	00000120
	REAL B	00000130
	REAL C	00000140
	REAL COEF1(30)	00000150
	REAL COEF2(30)	00000160
	REAL COEF3(30)	00000170
	REAL COEF4(30)	00000180
	REAL COEF5(30)	00000190
	REAL COEF6(30)	00000200
	REAL COEF7(30)	00000210
	REAL DATA(105)	00000220
	REAL DELAD(3)	00000230
	REAL DELVD(3)	00000240
	REAL DELWD(3)	00000250
	REAL DELVBP(3)	00000260
	REAL DELWBP(3)	00000270
	REAL DT	00000280
	REAL E	00000290
	REAL E1	00000300
	REAL ETA	00000310
	REAL F	00000320
	REAL FA	00000330
	REAL FY	00000340
	REAL G	00000350
	REAL GAS	00000360
	REAL GAUSS	00000370
	REAL GUST(6)	00000380
	REAL H	00000390
	REAL MOOPOT	00000400
	REAL P	00000410
	REAL P11	00000420
	REAL P12	00000430
	REAL P21	00000440
	REAL P22	00000450
	REAL PARMPI(30)	00000460
	REAL PARMQI(30)	00000470
	REAL PARMRI(30)	00000480
	REAL PCUF(16)	00000490
	REAL PDATA(20)	00000500
	REAL PI	00000510
	REAL FRNTDT	00000520
	REAL QBD(9)	00000530
	REAL QDCCKG(9)	00000540
	REAL QDBPSV(9)	00000550
	REAL QTEMP(9)	00000560
	REAL QTG(9)	00000570
	REAL R	00000580
	REAL RAND(6)	00000590
	REAL RE	00000600

REAL S(6)	00000610
REAL SA	00000620
REAL SPARE1	00000630
REAL T	00000640
REAL TPRMOD	00000650
REAL TPRNT	00000660
REAL TEMP	00000670
REAL TENV	00000680
REAL U	00000690
REAL V	00000700
REAL WB(3)	00000710
REAL WBB(3)	00000720
REAL WBBOLD(3)	00000730
REAL WBBDOT(3)	00000740
REAL WE	00000750
REAL WH	00000760
REAL WN	00000770
REAL WO	00000780
REAL WR	00000790
REAL WTEMP(9)	00000800
REAL X1(5)	00000810
REAL X1PRV(30)	00000820
REAL X2(5)	00000830
REAL X2PRV(30)	00000840
REAL Y	00000850
REAL Z	00000860
REAL ZETA	00000870
C	
INTEGER IENDF	00000880
INTEGER INITSW	00000890
INTEGER IX	00000900
INTEGER OUTSW	00000910
INTEGER IFILE	00000920
INTEGER PFILE	00000930
INTEGER OFILE	00000940
C	
EQUIVALENCE (PDATA(1), WF)	00000950
EQUIVALENCE (PDATA(2), RE)	00000960
EQUIVALENCE (PDATA(3), G)	00000970
EQUIVALENCE (PDATA(4), PRNTDT)	00000980
EQUIVALENCE (PDATA(5), PBUF(1))	00000990
C	
EQUIVALENCE (DATA(1), DT)	00001000
EQUIVALENCE (DATA(2), PRNTSW)	00001010
EQUIVALENCE (DATA(3), OUTSW)	00001020
EQUIVALENCE (DATA(4), XFILE)	00001030
EQUIVALENCE (DATA(5), S(1))	00001040
EQUIVALENCE (DATA(6), S(2))	00001050
EQUIVALENCE (DATA(7), S(3))	00001060
EQUIVALENCE (DATA(8), S(4))	00001070
EQUIVALENCE (DATA(9), S(5))	00001080
EQUIVALENCE (DATA(10), S(6))	00001090
EQUIVALENCE (DATA(11), PARMAP(1))	00001100
EQUIVALENCE (DATA(12), PARMAP(2))	00001110
EQUIVALENCE (DATA(13), PARMAP(3))	00001120
EQUIVALENCE (DATA(14), PARMAP(4))	00001130
EQUIVALENCE (DATA(15), PARMAP(5))	00001140
EQUIVALENCE (DATA(16), PARMAP(6))	00001150
EQUIVALENCE (DATA(17), PARMAP(7))	00001160
EQUIVALENCE (DATA(18), PARMAP(8))	00001170
	00001180
	00001190
	00001200

EQUIVALENCE (DATA(19), PARMAP(9))	00001210
EQUIVALENCE (DATA(20), PARMAP(10))	00001220
EQUIVALENCE (DATA(21), PARMAP(11))	00001230
EQUIVALENCE (DATA(22), PARMAP(12))	00001240
EQUIVALENCE (DATA(23), PARMAP(13))	00001250
EQUIVALENCE (DATA(24), PARMAP(14))	00001260
EQUIVALENCE (DATA(25), PARMAP(15))	00001270
EQUIVALENCE (DATA(26), PARMAP(16))	00001280
EQUIVALENCE (DATA(27), PARMAP(17))	00001290
EQUIVALENCE (DATA(28), PARMAP(18))	00001300
EQUIVALENCE (DATA(29), PARMAP(19))	00001310
EQUIVALENCE (DATA(30), PARMAP(20))	00001320
EQUIVALENCE (DATA(31), PARMAP(21))	00001330
EQUIVALENCE (DATA(32), PARMAP(22))	00001340
EQUIVALENCE (DATA(33), PARMAP(23))	00001350
EQUIVALENCE (DATA(34), PARMAP(24))	00001360
EQUIVALENCE (DATA(35), PARMAP(25))	00001370
EQUIVALENCE (DATA(36), PARMAP(26))	00001380
EQUIVALENCE (DATA(37), PARMAP(27))	00001390
EQUIVALENCE (DATA(38), PARMAP(28))	00001400
EQUIVALENCE (DATA(39), PARMAP(29))	00001410
EQUIVALENCE (DATA(40), PARMAP(30))	00001420
EQUIVALENCE (DATA(41), PARMWH(1))	00001430
EQUIVALENCE (DATA(42), PARMWH(2))	00001440
EQUIVALENCE (DATA(43), PARMWH(3))	00001450
EQUIVALENCE (DATA(44), PARMWH(4))	00001460
EQUIVALENCE (DATA(45), PARMWH(5))	00001470
EQUIVALENCE (DATA(46), PARMWH(6))	00001480
EQUIVALENCE (DATA(47), PARMWH(7))	00001490
EQUIVALENCE (DATA(48), PARMWH(8))	00001500
EQUIVALENCE (DATA(49), PARMWH(9))	00001510
EQUIVALENCE (DATA(50), PARMWH(10))	00001520
EQUIVALENCE (DATA(51), PARMWH(11))	00001530
EQUIVALENCE (DATA(52), PARMWH(12))	00001540
EQUIVALENCE (DATA(53), PARMWH(13))	00001550
EQUIVALENCE (DATA(54), PARMWH(14))	00001560
EQUIVALENCE (DATA(55), PARMWH(15))	00001570
EQUIVALENCE (DATA(56), PARMWH(16))	00001580
EQUIVALENCE (DATA(57), PARMWH(17))	00001590
EQUIVALENCE (DATA(58), PARMWH(18))	00001600
EQUIVALENCE (DATA(59), PARMWH(19))	00001610
EQUIVALENCE (DATA(60), PARMWH(20))	00001620
EQUIVALENCE (DATA(61), PARMWH(21))	00001630
EQUIVALENCE (DATA(62), PARMWH(22))	00001640
EQUIVALENCE (DATA(63), PARMWH(23))	00001650
EQUIVALENCE (DATA(64), PARMWH(24))	00001660
EQUIVALENCE (DATA(65), PARMWH(25))	00001670
EQUIVALENCE (DATA(66), PARMWH(26))	00001680
EQUIVALENCE (DATA(67), PARMWH(27))	00001690
EQUIVALENCE (DATA(68), PARMWH(28))	00001700
EQUIVALENCE (DATA(69), PARMWH(29))	00001710
EQUIVALENCE (DATA(70), PARMWH(30))	00001720
EQUIVALENCE (DATA(71), PARMWO(1))	00001730
EQUIVALENCE (DATA(72), PARMWO(2))	00001740
EQUIVALENCE (DATA(73), PARMWO(3))	00001750
EQUIVALENCE (DATA(74), PARMWO(4))	00001760
EQUIVALENCE (DATA(75), PARMWO(5))	00001770
EQUIVALENCE (DATA(76), PARMWO(6))	00001780
EQUIVALENCE (DATA(77), PARMWO(7))	00001790
EQUIVALENCE (DATA(78), PARMWO(8))	00001800

	EQUIVALENCE (DATA(79), PARMWO(9))	00001810
	EQUIVALENCE (DATA(80), PARMWO(10))	00001820
	EQUIVALENCE (DATA(81), PARMWO(11))	00001830
	EQUIVALENCE (DATA(82), PARMWO(12))	00001840
	EQUIVALENCE (DATA(83), PARMWO(13))	00001850
	EQUIVALENCE (DATA(84), PARMWO(14))	00001860
	EQUIVALENCE (DATA(85), PARMWO(15))	00001870
	EQUIVALENCE (DATA(86), PARMWO(16))	00001880
	EQUIVALENCE (DATA(87), PARMWO(17))	00001890
	EQUIVALENCE (DATA(88), PARMWO(18))	00001900
	EQUIVALENCE (DATA(89), PARMWO(19))	00001910
	EQUIVALENCE (DATA(90), PARMWO(20))	00001920
	EQUIVALENCE (DATA(91), PARMWO(21))	00001930
	EQUIVALENCE (DATA(92), PARMWO(22))	00001940
	EQUIVALENCE (DATA(93), PARMWO(23))	00001950
	EQUIVALENCE (DATA(94), PARMWO(24))	00001960
	EQUIVALENCE (DATA(95), PARMWO(25))	00001970
	EQUIVALENCE (DATA(96), PARMWO(26))	00001980
	EQUIVALENCE (DATA(97), PARMWO(27))	00001990
	EQUIVALENCE (DATA(98), PARMWO(28))	00002000
	EQUIVALENCE (DATA(99), PARMWO(29))	00002010
	EQUIVALENCE (DATA(100), PARMWO(30))	00002020
C		00002030
	EQUIVALENCE (DATA(101), MODPDT)	00002040
	EQUIVALENCE (DATA(102), VIBSW)	00002050
	EQUIVALENCE (DATA(103), GUSTLA)	00002060
	EQUIVALENCE (DATA(104), GUSTLO)	00002070
	EQUIVALENCE (DATA(105), GUSTNR)	00002080
C		00002090
	DATA DELWBP/3*0./	00002100
	DATA DELVBP/3*0./	00002110
	DATA PI/3.1415926535897/	00002120
	DATA QSBPRV/1.0,3*0.0,1.0,3*0.0,1.0/	00002130
	DATA RAND/6*0.0/	00002140
	DATA TPRMOD/0.0/	00002150
	DATA TPRNT/0.0/	00002160
	DATA INITSW/0/	00002170
	DATA IN1 /0/	00002180
	DATA IN2 /0/	00002190
	DATA IFILE/30/	00002200
	DATA PFILE/7/	00002210
C		00002220
	IF (IENDF.EQ.1) RETURN	00002230
	IF (INITSW.EQ.0) GO TO 500	00002240
	IF (T.LT.TENV-.0005) RETURN	00002250
	100 CONTINUE	00002260
C		00002270
C		00002280
C	IF VIBRATION SWITCH IS ZERO BYPASS VIBRATION COMPUTATION	00002290
C	AND ADD ZERO RATE AND ACCELATION	00002300
C		00002310
	IF (VIBSW.LT.1) GO TO 949	00002320
C		00002330
C	THE FOLLOWING GENERATES SEQUENCES OF RANDOM NUMBERS	00002340
C	X(VIBRATIONS) CORRESPONDING TO DESIGNED PSD'S	00002350
C		00002360
C		00002370
	00 940 J=1,6,1	00002380
C		00002390
C		00002400

	Y=0.0	00002410
	N=S(J)	00002420
	DO 930 I=1,N,1	00002430
C	ETA=GAUSS (0.0,1.0)	00002440
	ZETA=GAUSS (0.0,1.0)	00002450
C		00002460
	K=5*(J-1) +I	00002470
C		00002480
C		00002490
	X1(I)=COEF6(K)*X1PRV(K) + COEF4(K)*X2PRV(K) +	00002500
	XCOEF2(K)*ETA + COEF1(K)*ZETA	00002510
	X2(I)=COEF7(K)*X1PRV(K) + COEF5(K)*X2PRV(K) +	00002520
	XCOEF3(K)*ETA	00002530
	IF(IN2.NE.0) GOTO 925	00002540
	Y=Y+X1(I)/DT	00002550
	GOTO 926	00002560
925	CONTINUE	00002570
	Y=Y+(X1(I)-X1PRV(K))/DT	00002580
926	CONTINUE	00002590
	X1PRV(K)=X1(I)	00002600
	X2PRV(K)=X2(I)	00002610
C		00002620
	930 CONTINUE	00002630
	RAND(J)=Y	00002640
	940 CONTINUE	00002650
C		00002660
	IN2=1	00002670
C		00002680
C	VIBRATION GENERATOR COMPLETE	00002690
C		00002700
C	COMPUTE ENVIRONMENT OUTPUT	00002710
C		00002720
	DO 950 I=1,3	00002730
	K=I+3	00002740
	WBBOLD(I)=WBB(I)	00002750
	DELVB(I)=RAND(I)	00002760
	DELAB(I)=(DELVB(I)-DELVBP(I))/DT	00002770
	DELVBP(I)=DELVB(I)	00002780
	DELWB(I)=RAND(K)	00002790
950	CONTINUE	00002800
	INI=1	00002810
	WBB(1)=WB(1)+DELWB(3)	00002820
	WBB(2)=WB(2)+DELWB(1)	00002830
	WBB(3)=WB(3)+DELWB(2)	00002840
C	COMPUTE THE CHANGE IN BODY ATTITUDE DUE TO VIBRATION	00002850
	QBBCHG(1)=1.	00002860
	QBBCHG(2)=(DELWB(2)+DELWBP(2))/2.*DT	00002870
	QBBCHG(3)=-((DELWB(1)+DELWBP(1))/2.*DT	00002880
	QBBCHG(4)=-((DELWB(2)+DELWBP(2))/2.*DT	00002890
	QBBCHG(5)=1.	00002900
	QBBCHG(6)=(DELWB(3)+DELWBP(3))/2.*DT	00002910
	QBBCHG(7)=(DELWB(1)+DELWBP(1))/2.*DT	00002920
	QBBCHG(8)=-((DELWB(3)+DELWBP(3))/2.*DT	00002930
	QBBCHG(9)=1.	00002940
	CALL MXM(QBBCHG,QBBPRV,QBB)	00002950
	DELWBP(1)=DELWB(1)	00002960
	DELWBP(2)=DELWB(2)	00002970
	DELWBP(3)=DELWB(3)	00002980
C	ORTHONORMALIZE THE ATTITUDE MATRIX	00002990
		00003000

CALL MTXM(QBB,QBB,QTQ)	00003010
DO 952 I4=1,9	00003020
QTQ(I4)=-.5*QTQ(I4)	00003030
952 CONTINUE	00003040
QTQ(1)=1.5*QTQ(1)	00003050
QTQ(5)=1.5*QTQ(5)	00003060
QTQ(9)=1.5*QTQ(9)	00003070
CALL MXM(QBB,QTQ,QTEMP)	00003080
DO 953 I5=1,9	00003090
QBB(I5)=QTEMP(I5)	00003100
953 CONTINUE	00003110
C END ORTHONORMALIZATION	00003120
C CALCULATE ROTATED ACCELERATION	00003130
CALL MXV(QBB,AB,ABB)	00003140
DO 954 I=1,3	00003150
WBDOT(I)=(WBB(I)-WBBOLD(I))/DT	00003160
954 CONTINUE	00003170
DO 955 I=1,9	00003180
QBBPRV(I)=QBB(I)	00003190
955 CONTINUE	00003200
ABB(1)=ABB(1)+DELAB(3)*G	00003210
ABB(2)=ABB(2)+DELAB(2)*G	00003220
ABB(3)=ABB(3)+DELAB(1)*G	00003230
GOTO 956	00003240
949 CONTINUE	00003250
DO 957 I=1,3	00003260
WBBOLD(I)=WBB(I)	00003270
WBB(I)=WB(I)	00003280
ABB(I)=AB(I)	00003290
WBDOT(I)=(WBB(I)-WBBOLD(I))/DT	00003300
957 CONTINUE	00003310
956 CONTINUE	00003320
C	00003330
C OUTPUT AND PRINT CONTROL	00003340
C	00003350
IF (PRNTDT.GT.0) GO TO 960	00003360
IF (MODPDT.EQ.0) GO TO 999	00003370
C	00003380
IF (T.LT.TPRMOD-.0005) GO TO 999	00003390
TPRMOD=TPRMOD+MODPDT	00003400
GO TO 970	00003410
C	00003420
960 CONTINUE	00003430
IF (T.LT.TPRNT-.0005) GO TO 999	00003440
TPRNT=TPRNT+PRNTDT	00003450
C	00003460
970 CONTINUE	00003470
IF (PRNTSW.LT.1) GO TO 999	00003480
C	00003490
WRITE(OFIL,1200) AB,WB,ABB,WBB,WBDOT	00003500
C	00003510
C	00003520
C	00003530
999 CONTINUE	00003540
TENV=T+DT	00003550
RETURN	00003560
C	00003570
C ENVIRONMENT INITIALIZATION BEGINS HERE	00003580
C	00003590
500 CONTINUE	00003600

REWIND IFILE	00003610
REWIND PFILE	00003620
501 READ (IFILE,1000) IX,DATA(IX)	00003630
IF (EOF(IFILE)) 502,501	00003640
502 CONTINUE	00003650
503 READ (PFILE,1000) IX,PDATA(IX)	00003660
IF (EOF(PFILE)) 510,503	00003670
510 CONTINUE	00003680
REWIND IFILE	00003690
REWIND PFILE	00003700
C	00003710
OFIL=XFILE	00003720
C	00003730
C	00003740
C THE FOLLOWING GENERATES INITIAL VALUES FOR RANDOM NUMBERS	00003750
C HAVING SPECIFIED TURBULENCE PSD'S	00003760
C	00003770
C GUST IS IN UNITS OF FT/SEC AND REPRESENTS VARIANCE OF WIND VELOCITY	00003780
GUST(1)=GUSTNR	00003790
GUST(2)=GUSTLA	00003800
GUST(3)=GUSTLO	00003810
GUST(4)=GUSTNR	00003820
GUST(5)=GUSTLA	00003830
GUST(6)=GUSTLA	00003840
C	00003850
L=1	00003860
C	00003870
DO 600 J=1,6,1	00003880
N=S(J)	00003890
IF (N.EQ.0) GO TO 600	00003900
DO 605 I=1,N,1	00003910
K=5*(J-1)+I	00003920
C	00003930
ETA=GAUSS (0.0,1.0)	00003940
ZETA=GAUSS (0.0,1.0)	00003950
C	00003960
AP=PARMAPI(K)	00003970
WH=PARMMH(K)	00003980
WO=PARMWO(K)	00003990
C	00004000
AP=GUST(J) *2*PI*AP	00004010
A=WH/WO	00004020
SA=A**2	00004030
TEMP=SQRT(2-(1-SA/2)**2)	00004040
WN=WO*SQRT(TEMP)	00004050
E=SQRT((1-(WO/WN)**2)/2)	00004060
F=AP*(1-(1-2*E**2)**2)	00004070
X1PRV(K)=0.	00004080
X2PRV(K)=0.	00004090
WR=WN*SQRT(1-E**2)	00004100
Z=E*WN	00004110
E1=EXP(-2*Z*DT)	00004120
C=ASIN(E)	00004130
C	00004140
U=(1-E1)/(4*Z*WR**2)+	00004150
X(E1*(Z*COS(2*WR*DT)-WR*SIN(2*WR*DT)))/(4*(WN**2)*(WR**2))	00004160
X=E/(4*WN*WR**2)	00004170
C	00004180
R=(1-E1)/((4*E*WR**2)/WN)-	00004190
X(E1*(Z*COS(2*WR*DT+2*C)-WR*SIN(2*WR*DT+2*C)))/(4*WR**2)	00004200

	X+(Z*COS(2*C)-WR*SIN(2*C))/(4*WR**2)	00004210
C	V=(E1-1)*SIN(C)/(4*E*WR**2)-	00004220
	X(E1*(Z*SIN(2*WR*DT+C)+WR*COS(2*WR*DT+C)))/(4*WN*WR**2)	00004230
	X+(WR*COS(C)+Z*SIN(C))/(4*WN*WR**2)	00004240
C	B=SQRT(ABS(U-(V**2)/R))*SQRT(F)*WN**2	00004250
	H=(V/SQRT(R))*SQRT(F)*WN*WN	00004260
	P=(SQRT(R))*SQRT(F)*WN**2	00004270
C	P12=(SIN(WR*DT)*EXP(-Z*DT))/WR	00004280
	P22=(COS(WR*DT)-(Z/WR)*SIN(WR*DT))*EXP(-Z*DT)	00004290
	P11=2*Z*P12+P22	00004300
	P21=(-WN**2)*P12	00004310
C	COEF1(K)=B	00004320
	COEF2(K)=H	00004330
	COEF3(K)=P	00004340
	COEF4(K)=P12	00004350
	COEF5(K)=P22	00004360
	COEF6(K)=P11	00004370
	COEF7(K)=P21	00004380
C	605 CONTINUE	00004390
	600 CONTINUE	00004400
C	PSD'S COEF PARAMETERS COMPUTED FOR A SET OF AMPLITUDE,	00004410
C	HALF-WIDTH, RESONANT FREQUENCY AND GUST	00004420
C	DO 610 I=1,3	00004430
	WDB(I)=WB(I)	00004440
	ABB(I)=AB(I)	00004450
	WBBDOT(I)=0.0	00004460
	610 CONTINUE	00004470
C	INITIALIZATION OUTPUT AND PRINT CONTROL	00004480
C	WRITE(OFIL,1010)DT,PRNTSW,OUTSW,OFIL,PRNTDT,MODPOT,VIBSW	00004490
C	700 CONTINUE	00004500
	N=S(1)	00004510
	WRITE(OFIL,1020) N	00004520
	J=1	00004530
	GO TO 750	00004540
701 CONTINUE		00004550
	N=S(2)	00004560
	WRITE(OFIL,1021) N	00004570
	J=2	00004580
	GO TO 750	00004590
702 CONTINUE		00004600
	N=S(3)	00004610
	WRITE(OFIL,1022) N	00004620
	J=3	00004630
	GO TO 750	00004640
703 CONTINUE		00004650
	N=S(4)	00004660
	WRITE(OFIL,1023) N	00004670
	J=4	00004680
	GO TO 750	00004690
704 CONTINUE		00004700
		00004710
		00004720
		00004730
		00004740
		00004750
		00004760
		00004770
		00004780
		00004790
		00004800

N=S(5)	00004810
WRITE(OFIL,1024) N	00004820
J=5	00004830
GO TO 750	00004840
705 CONTINUE	00004850
N=S(6)	00004860
WRITE(OFIL,1025) N	00004870
J=6	00004880
GO TO 750	00004890
706 CONTINUE	00004900
WRITE(OFIL,1026)	00004910
C	00004920
INITSW=1	00004930
TENV=T+DT	00004940
RETURN	00004950
C	00004960
C	00004970
750 CONTINUE	00004980
IF (N.EQ.0) GO TO 752	00004990
DO 751 I=1,N,1	00005000
K=5*(J-1)+I	00005010
WRITE(OFIL,1030)PARMAF(K),PARMWH(K),PARMWO(K)	00005020
751 CONTINUE	00005030
752 GO TO (701,702,703,704,705,706),J	00005040
C	00005050
C	00005060
1000 FORMAT (I5,F20.10)	00005070
1010 FORMAT(//,30H ENVIRONMENT INITIALIZATION ,	00005080
X/3X,8H DT ,3X,G16.8,3X,4H SEC,	00005090
X/3X,8H FRNTSW ,3X,G16.8,	00005100
X/3X,8H OUTSW ,3X,G16.8,	00005110
X/3X,8H OFIL ,3X,I15,	00005120
X/3X,8H PRNTDT ,3X,G16.8,	00005130
X/3X,8H MODPDT ,3X,G16.8,	00005140
X/3X,8H VIBSW ,3X,G16.8,//)	00005150
C	00005160
1020 FORMAT(3X,32H VIBRATION GENERATOR PARAMETERS ,//,	00005170
X 3X,20H VERTICAL LOAD ,14X,4H AP ,8X,4H WH ,6X,4H WD ,	00005180
X /6X,I4,7H PEAKS)	00005190
1021 FORMAT(/3X,20H LATERAL LOAD ,	00005200
X /6X,I4,7H PEAKS)	00005210
1022 FORMAT(/3X,20H LONGITUDINAL LOAD ,	00005220
X /6X,I4,7H PEAKS)	00005230
1023 FORMAT(/3X,20H PITCH RATE ,	00005240
X /6X,I4,7H PEAKS)	00005250
1024 FORMAT(/3X,20H YAW RATE PAD ,	00005260
X /6X,I4,7H PEAKS)	00005270
1025 FORMAT(/3X,20H ROLL RATE PSD ,	00005280
X /6X,I4,7H PEAKS)	00005290
1026 FORMAT(//)	00005300
C	00005310
1030 FORMAT(27X,G16.8,5X,G16.8,7X,G16.8)	00005320
1100 FORMAT(1H+,8X,G16.8)	00005330
1200 FORMAT(6X,25H ** ENV ** AB(FT/SEC2) ,3G16.8,/	00005340
X 6X,25H WB(RAD/SEC) ,3G16.8,/	00005350
X 6X,25H ABB(FT/SEC2) ,3G16.8,/	00005360
X 6X,25H WDB(RAD/SEC) ,3G16.8,/	00005370
X 6X,25H WBBDOT(R/S2) ,3G16.8,//)	00005380
C	00005390
C	00005400

1300 FORMAT(6X,6H TEMP , 3G16.8,/12X,3G16.8,/12X,3G16.8,/)
1400 FORMAT(3X,3F16.8,/)
1410 FORMAT(3X,7F9.6,/)
END

00005410
00005420
00005430
00005440

C		00000010
C	03/10/78 DATE OF CURRENT MODULE	00000020
C		00000030
C	THE ACCELEROMETER MODULE SIMULATES A PENDULOUS SINGLE DEGREE OF	00000040
C	FREEDOM (SDF) FLOATED ACCELEROMETER.	00000050
C		00000060
	SUBROUTINE ACCEL(T,IENDF,ABB,WBB,WOOT,	00000070
	X DV)	00000080
C		00000090
	REAL ABA(3)	00000100
	REAL ABB(3)	00000110
	REAL ABC(3)	00000120
	REAL ABIAS(3)	00000130
	REAL AQUANT	00000140
	REAL A20	00000150
	REAL SIA	00000160
	REAL BIAS(3)	00000170
	REAL BIASA(3)	00000180
	REAL BIASV(3)	00000190
	REAL CO	00000200
	REAL CROSS1(3)	00000210
	REAL CROSS2(3)	00000220
	REAL DATA(97)	00000230
	REAL DELT	00000240
	REAL DEN	00000250
	REAL DT	00000260
	REAL DTHETI(3)	00000270
	REAL DTI	00000280
	REAL DV(3)	00000290
	REAL EX	00000300
	REAL G	00000310
	REAL GAUSS	00000320
	REAL HT	00000330
	REAL I	00000340
	REAL LAT	00000350
	REAL K	00000360
	REAL KO(3)	00000370
	REAL KP(3)	00000380
	REAL KII(3)	00000390
	REAL KPP(3)	00000400
	REAL KIO(3)	00000410
	REAL KOP(3)	00000420
	REAL KIP(3)	00000430
	REAL MODPDT	00000440
	REAL MRC	00000450
	REAL M1	00000460
	REAL M2	00000470
	REAL ORDER	00000480
	REAL PBUF(16)	00000490
	REAL PDATA(20)	00000500
	REAL PRNTDT	00000510
	REAL QBAX(9)	00000520
	REAL QBAY(9)	00000530
	REAL QBAZ(9)	00000540
	REAL QUANT	00000550
	REAL RE	00000560
	REAL RX(3)	00000570
	REAL RY(3)	00000580
	REAL RZ(3)	00000590
	REAL SFMO(3)	00000600

REAL SFM1(3)	00000610
REAL SFP0(3)	00000620
REAL SFP1(3)	00000630
REAL SM0(3)	00000640
REAL SM1(3)	00000650
REAL SP0(3)	00000660
REAL SP1(3)	00000670
REAL SPARE1	00000680
REAL SPARE2	00000690
REAL T	00000700
REAL TACC	00000710
REAL TDTHET	00000720
REAL THOOT(3)	00000730
REAL THETA(3)	00000740
REAL THN	00000750
REAL TPRMOD	00000760
REAL TFRNT	00000770
REAL TT	00000780
REAL TTTHET(3)	00000790
REAL WBA(3)	00000800
REAL WBB(3)	00000810
REAL WDOT(3)	00000820
REAL WJOTA(3)	00000830
REAL WE	00000840
C	00000850
INTEGER IENDF	00000860
INTEGER INITSW	00000870
INTEGER IFILE	00000880
INTEGER OFILE	00000890
INTEGER OUTSW	00000900
INTEGER PFILE	00000910
C	00000920
EQUIVALENCE (DATA(1), DT)	00000930
EQUIVALENCE (DATA(2), PRNTSW)	00000940
EQUIVALENCE (DATA(3), OUTSW)	00000950
EQUIVALENCE (DATA(4), XFILE)	00000960
EQUIVALENCE (DATA(5), SPARE1)	00000970
EQUIVALENCE (DATA(6), SPARE2)	00000980
EQUIVALENCE (DATA(7), CO)	00000990
EQUIVALENCE (DATA(13), QBAX(1))	00001000
EQUIVALENCE (DATA(22), QBAY(1))	00001010
EQUIVALENCE (DATA(31), QBAZ(1))	00001020
EQUIVALENCE (DATA(40), MRC)	00001030
EQUIVALENCE (DATA(42), QUANT)	00001040
EQUIVALENCE (DATA(43), BIAS(1))	00001050
EQUIVALENCE (DATA(46), K)	00001060
EQUIVALENCE (DATA(47), I)	00001070
EQUIVALENCE (DATA(48), DELI)	00001080
EQUIVALENCE (DATA(51), KO(1))	00001090
EQUIVALENCE (DATA(54), KP(1))	00001100
EQUIVALENCE (DATA(57), KII(1))	00001110
EQUIVALENCE (DATA(60), KPP(1))	00001120
EQUIVALENCE (DATA(63), KIO(1))	00001130
EQUIVALENCE (DATA(66), KIP(1))	00001140
EQUIVALENCE (DATA(69), KOP(1))	00001150
EQUIVALENCE (DATA(72), BIASV(1))	00001160
EQUIVALENCE (DATA(75), SFP0(1))	00001170
EQUIVALENCE (DATA(78), SFM0(1))	00001180
EQUIVALENCE (DATA(81), MODPDT)	00001190
EQUIVALENCE (DATA(82), ORDER)	00001200

	EQUIVALENCE (DATA(83),SFPI(1))	00001210
	EQUIVALENCE (DATA(86),SFMI(1))	00001220
	EQUIVALENCE (DATA(89),RX(1))	00001230
	EQUIVALENCE (DATA(92),RY(1))	00001240
	EQUIVALENCE (DATA(95),RZ(1))	00001250
C		00001260
	EQUIVALENCE (PDATA(1), WE)	00001270
	EQUIVALENCE (PDATA(2), RE)	00001280
	EQUIVALENCE (PDATA(3), G)	00001290
	EQUIVALENCE (PDATA(4), PRNTDT)	00001300
	EQUIVALENCE (PDATA(5), PBUF(1))	00001310
C		00001320
	DATA ABIAS /3*0./	00001330
	DATA AM /0./	00001340
	DATA THDOT/3*0./	00001350
	DATA THETA/3*0./	00001360
	DATA TPRMOD/0.0/	00001370
	DATA TPRNT/0.0/	00001380
	DATA TTHET/3*0./	00001390
	DATA INITSW/0/	00001400
	DATA IFILE/50/	00001410
	DATA ITHET/0/	00001420
	DATA IX/3/	00001430
	DATA K10/0/	00001440
	DATA K11/0/	00001450
	DATA K12/0/	00001460
	DATA PFILE/7/	00001470
C		00001480
	IF (IENDF.EQ.1) RETURN	00001490
	IF (INITSW.EQ.0) GO TO 500	00001500
	IF (T.LT.TACC-.0001) RETURN	00001510
C		00001520
	K10=K10+1	00001530
	DO 400 I1=1, 3	00001540
	IF (I1.GT.1) GOTO 110	00001550
C		00001560
C	TRANSFORM FROM BODY TO ACCELEROMETER COORDINATES(FROM XYZ TO IOP)	00001570
C		00001580
C	TRANSFORM X ACCELEROMETER	00001590
C	CALCULATE LEVER ARM EFFECT	00001600
C		00001610
	CROSS1(1)=WBB(2)*RX(3)-WBB(3)*RX(2)	00001620
	CROSS1(2)=WBB(3)*RX(1)-WBB(1)*RX(3)	00001630
	CROSS1(3)=WBB(1)*RX(2)-WBB(2)*RX(1)	00001640
	CROSS2(1)=WBB(2)*CROSS1(3)-WBB(3)*CROSS1(2)	00001650
	CROSS2(2)=WBB(3)*CROSS1(1)-WBB(1)*CROSS1(3)	00001660
	CROSS2(3)=WBB(1)*CROSS1(2)-WBB(2)*CROSS1(1)	00001670
	CROSS1(1)=WDOT(2)*RX(3)-WDOT(3)*RX(2)	00001680
	CROSS1(2)=-WDOT(1)*RX(3)+WDOT(3)*RX(1)	00001690
	CROSS1(3)=WDOT(1)*RX(2)-WDOT(2)*RX(1)	00001700
C		00001710
	DO 105 I2=1,3	00001720
	ABC(I2)=ABB(I2)+CROSS2(I2)+CROSS1(I2)	00001730
105	CONTINUE	00001740
C		00001750
	CALL MXV(QDAX,ABC,ABA)	00001760
	CALL MXV(QDAX,WBB,WBA)	00001770
	CALL MXV(QDAX,WDOT,WDOTA)	00001780
	GOTO 130	00001790
C		00001800

110	IF(I1.GT.2) GOTO 120	00001810
C		00001820
C	TRANSFORM Y ACCELEROMETER	00001830
C	CALCULATE LEVER ARM EFFECT	00001840
C		00001850
	CROSS1(1)=WBB(2)*RY(3)-WBB(3)*RY(2)	00001860
	CROSS1(2)=-WBB(1)*RY(3)+WBB(3)*RY(1)	00001870
	CROSS1(3)=WBB(1)*RY(2)-WBB(2)*RY(1)	00001880
	CROSS2(1)=WBB(2)*CROSS1(3)-WBB(3)*CROSS1(2)	00001890
	CROSS2(2)=WBB(3)*CROSS1(1)-WBB(1)*CROSS1(3)	00001900
	CROSS2(3)=WBB(1)*CROSS1(2)-WBB(2)*CROSS1(1)	00001910
	CROSS1(1)=WDOT(2)*RY(3)-WDOT(3)*RY(2)	00001920
	CROSS1(2)=-WDOT(1)*RY(3)+WDOT(3)*RY(1)	00001930
	CROSS1(3)=WDOT(1)*RY(2)-WDOT(2)*RY(1)	00001940
C		00001950
	DO 115 I2=1,3	00001960
	ABC(I2)=ABB(I2)+CROSS2(I2)+CROSS1(I2)	00001970
115	CONTINUE	00001980
	CALL MXV(QBAY,ABC,ABA)	00001990
	CALL MXV(QBAY,WBB,WBA)	00002000
	CALL MXV(QBAY,WDOT,WDOTA)	00002010
	GOTO 130	00002020
C		00002030
C	TRANSFORM Z ACCELEROMETER	00002040
C	CALCULATE LEVER ARM EFFECT	00002050
C		00002060
120	CROSS1(1)=WBB(2)*RZ(3)-WBB(3)*RZ(2)	00002070
	CROSS1(2)=-WBB(1)*RZ(3)+WBB(3)*RZ(1)	00002080
	CROSS1(3)=WBB(1)*RZ(2)-WBB(2)*RZ(1)	00002090
	CROSS2(1)=WBB(2)*CROSS1(3)-WBB(3)*CROSS1(2)	00002100
	CROSS2(2)=WBB(3)*CROSS1(1)-WBB(1)*CROSS1(3)	00002110
	CROSS2(3)=WBB(1)*CROSS1(2)-WBB(2)*CROSS1(1)	00002120
	CROSS1(1)=WDOT(2)*RZ(3)-WDOT(3)*RZ(2)	00002130
	CROSS1(2)=-WDOT(1)*RZ(3)+WDOT(3)*RZ(1)	00002140
	CROSS1(3)=WDOT(1)*RZ(2)-WDOT(2)*RZ(1)	00002150
C		00002160
	DO 125 I2=1,3	00002170
	ABC(I2)=ABB(I2)+CROSS2(I2)+CROSS1(I2)	00002180
125	CONTINUE	00002190
C		00002200
	CALL MXV(QBAZ,ABC,ABA)	00002210
	CALL MXV(QBAZ,WBB,WBA)	00002220
	CALL MXV(QBAZ,WDOT,WDOTA)	00002230
C		00002240
130	CONTINUE	00002250
C	PRINT 994,CROSS2,CROSS1	00002260
994	FORMAT (1X,3E15.7)	00002270
C		00002280
	IF (K10.NE.1) GOTO 135	00002290
	THETA(I1)=PI*C/K*ABA(1)	00002300
	WDOTA(2)=0.00	00002310
C		00002320
135	CONTINUE	00002330
	IF (K11.EQ.0) GOTO 138	00002340
	BI=BIASA(I1)	00002350
C		00002360
C	CALCULATE EXPONENTIALLY CORRELATED RANDOM BIAS	00002370
C		00002380
	ABIAS(I1)=ABIAS(I1)*EX*GAUSSIAN,BI)	00002390
	BIA=BIAS(I1)+ABIAS(I1)	00002400

C		00002410
	GOTO 139	00002420
139	BIA=BIAS(I1)	00002430
139	CONTINUE	00002440
C		00002450
C	CALCULATE THETA	00002460
C		00002470
	DEN=K*DELI*(WBA(3)*WBA(3)-WBA(1)*WBA(1)+ABA(3)*MRC	00002480
	M1 = MRC*(ABA(1)-BIA)+DELI*WBA(3)*WBA(1)-I*WDOTA(2)	00002490
C		00002500
C	IF ACCELERATION AND ACCELERATION**2 TERMS ARE ZERO, BRANCH AROUND	00002510
	M2=0.0	00002520
	IF (K12.NE.1) GOTO 850	00002530
	M2 = MRC*(-KP(I1)*ABA(3)-KO(I1)*ABA(2)	00002540
	X -KPP(I1)*ABA(3)*ABA(3)-KII(I1)*ABA(1)*ABA(1)-KIP(I1)*ABA(1)	00002550
	X *ABA(3)-KIO(I1)*ABA(1)*ABA(2)-KOP(I1)*ABA(2)*ABA(3))	00002560
850	CONTINUE	00002570
C		00002580
	IF (NORDER.NE.0) GOTO 700	00002590
C	PERFORMANCE MODEL	00002600
C		00002610
	THN=(M1+M2)/DEN	00002620
	GOTO 800	00002630
C		00002640
700	IF (NORDER.NE.1) GOTO 750	00002650
C	FIRST ORDER DIFFERENTIAL EQUATIONMODEL	00002660
C		00002670
	THN=THETA(I1)*DT*(-DEN*THETA(I1)+M1+M2)/CO	00002680
	GOTO 800	00002690
C		00002700
750	IF (NORDER.NE.2) GOTO 800	00002710
C	SECOND ORDER DIFFERENTIAL EQUATIONMODEL	00002720
C		00002730
	THN=THETA(I1)*DT*THDOT(I1)	00002740
	THDOT(I1)=THDOT(I1)*(-CO*THDOT(I1)-DEN*THETA(I1)+M1+M2)*DT	00002750
800	CONTINUE	00002760
C		00002770
C	CALCULATE QUANTIZED OUTPUT	00002780
	YTHET(I1)=THN*YTHET(I1)	00002790
	IF (YTHET(I1).LT.0.) GOTO 140	00002800
	T1=SP0(I1)+1.+SP1(I1)*THN*K/MRC	00002810
	GOTO 145	00002820
140	T1=SH0(I1)+1.+SH1(I1)*THN*K/MRC	00002830
145	YTHET=YTHET(I1)/AQUNT/T1	00002840
	YT=YTHET	00002850
	YTHET(I1)=YTHET(I1)-YT*T1*AQUANT	00002860
	THETA(I1)=THN	00002870
C	INTEGRATE BY	00002880
	DV(I1)=DV(I1)+YT*AQUANT/MRC*K*DT	00002890
C		00002900
C	PRINT 151,THN,THDOT(I1),YTHET,YTHET(I1),ABA(1),WBA(1),THETA(I1)	00002910
151	FORMAT (1X,2E15.6,110.7,4E15.7)	00002920
400	CONTINUE	00002930
C		00002940
C	OUTPUT AND PRINT CONTROL	00002950
C		00002960
	IF (PRINTOT.GT.0) GO TO 960	00002970
	IF (INCPOT.EQ.0) GO TO 999	00002980
C		00002990
	IF (T.LT.TPRKND-.0005) GO TO 999	00003000

TPRMOD=TPRMOD+MODPDT	00003010
GO TO 970	00003020
C	00003030
960 CONTINUE	00003040
IF (T.LT.TPRNT-.0005) GO TO 999	00003050
TPRNT=TPRNT+PRNTDT	00003060
C	00003070
970 CONTINUE	00003080
IF (PRNTSW.LT.1.) GO TO 999	00003090
WRITE(OFIL,1200) DV	00003100
C	00003110
C	00003120
999 CONTINUE	00003130
TACC=T+DT	00003140
RETURN	00003150
C	00003160
C	00003170
2000 WRITE (OFIL,2010)	00003180
2010 FORMAT ("ORDER NOT PROPERLY SPECIFIED")	00003190
STOP	00003200
C	00003210
C ACCELEROMETER INITIALIZATION BEGINS HERE	00003220
C	00003230
500 CONTINUE	00003240
REWIND IFILE	00003250
REWIND FFILE	00003260
501 READ (IFIL,1000) IQ,DATA(IQ)	00003270
IF (EQF(IFIL)) 502,501	00003280
502 CONTINUE	00003290
503 READ (FFIL,1000) IQ,PDATA(IQ)	00003300
IF (EQF(FFIL)) 510,503	00003310
510 CONTINUE	00003320
C	00003330
REWIND IFILE	00003340
REWIND FFILE	00003350
OFIL=XFIL	00003360
C	00003370
DO 519 J1=1,3	00003380
DV(J1)=0.0	00003390
SF0(J1)=SF0(J1)*1.E-6	00003400
SP1(J1)=SP1(J1)*1.E-6/G	00003410
SM0(J1)=SM0(J1)*1.E-6	00003420
SM1(J1)=SM1(J1)*1.E-6/G	00003430
519 CONTINUE	00003440
C	00003450
AQUANT=QUANT*MRC/K/30.48/DT	00003460
EX=EXP(-DT/40)	00003470
NORDER=ORDER	00003480
C	00003490
WRITE (OFIL,1010) DT,PRNTSW,MODPDT,PRNTDT,OUTSW,OFIL,	00003500
X (QBAX(I2),I2=1,9),	00003510
X (QBAY(I2),I2=1,9),(QBAZ(I2),I2=1,9)	00003520
WRITE(OFIL,1011) I,DELI,CO,MRC,QUANT	00003530
WRITE (OFIL,1012) K, KO,KP,KIT,KPP,KIO,	00003540
X KIP,KOP,BIAS,BIASV,SF0,SFM0,SP1,SM1,RX,RY,RZ,NORDER	00003550
C	00003560
DO 505 I0=1,3	00003570
BIASA(I0)=SQRT(BIASV(I0)*(1-EX*EX))*1.E-6*G	00003580
BIAS(I0)=BIASA(I0)*1.E-6*G	00003590
IF (BIASV(I0).NE.0.) K11=1	00003600

525 CONTINUE

C

X2=1.E-6/G
DO 530 I7=1,3

C

IF (KO(I7).NE.0.) K12=1
KO(I7)=KO(I7)*1.E-6
IF (KP(I7).NE.0.) K12=1
KP(I7)=KP(I7)*1.E-6
IF (KII(I7).NE.0.) K12=1
KII(I7)=KII(I7)*X2
IF (KIO(I7).NE.0.) K12=1
KIO(I7)=KIO(I7)*X2
IF (KIP(I7).NE.0.) K12=1
KIP(I7)=KIP(I7)*X2
IF (KOP(I7).NE.0.) K12=1
KOP(I7)=KOP(I7)*X2
IF (KPP(I7).NE.0.) K12=1
KPP(I7)=KPP(I7)*X2

530

CONTINUE

C

MRC=MRC*7.23E-5
CO=CO*2.37E-6
DELI=DELI*2.37E-6
K=K*2.37E-6
I=I*2.37E-6
DTI=DT/I

C

INITSW=1
TACC=T+DT
RETURN

C

C

1000 FORMAT (15,1X,F20.10)

1010 FORMAT(30H ACCELEROMETER INITIALIZATION

X/3X,0H DT .3X,G16.8,3X,4H SEC,

X/3X,0H PRNTSW .3X,G16.8,

X/3X,0H MODPST .3X,G16.8,

X/3X,0H PRNTDT .3X,G16.8,

X/3X,0H OUTSW .3X,G16.8,

X/3X,0H OFFILE .3X,I15,

X/3X,0H QDAX = .3(3X,G16.8),

X/9X,3(3X,G16.8),

X/9X,3(3X,G16.8),

X/3X,0H QBAY = .3(3X,G16.8),

X/9X,3(3X,G16.8),

X/9X,3(3X,G16.8),

X/3X,0H QDAZ = .3(3X,G16.8),

X/9X,3(3X,G16.8),

X/9X,3(3X,G16.8))

1011 FORMAT(

X/3X,0H I .3X,G16.8,0H GN CH**2,

X/3X,0H DELI .3X,G16.8,0H GN CH**2,

X/3X,0H CO .3X,G16.8,16H DYNE CM/RAD/SEC,

X/3X,0H MRC .3X,G16.8,0H GN CM,

X/3X,0H CUANT .3X,G16.8,7H CM/SEC)

1012 FORMAT (3X,0H K .3X,G16.8,30H GN CH**2/RAD SEC**2,

X/21X,"X",10X,"Y",16X,"Z",

X/3X,0H KO .3(3X,G16.8),10H MICRO G/G,

X/3X,0H KP .3(3X,G16.8),10H MICRO G/G,

00003610
00003620
00003630
00003640
00003650
00003660
00003670
00003680
00003690
00003700
00003710
00003720
00003730
00003740
00003750
00003760
00003770
00003780
00003790
00003800
00003810
00003820
00003830
00003840
00003850
00003860
00003870
00003880
00003890
00003900
00003910
00003920
00003930
00003940
00003950
00003960
00003970
00003980
00003990
00004000
00004010
00004020
00004030
00004040
00004050
00004060
00004070
00004080
00004090
00004100
00004110
00004120
00004130
00004140
00004150
00004160
00004170
00004180
00004190
00004200

X/3X,8H KII	,3(3X,G16.8),13H MICRO G/G**2,	00004210
X/3X,8H KPP	,3(3X,G16.8),13H MICRO G/G**2,	00004220
X/3X,8H KIO	,3(3X,G16.8),13H MICRO G/G**2,	00004230
X/3X,8H KIP	,3(3X,G16.8),13H MICRO G/G**2,	00004240
X/3X,8H KOP	,3(3X,G16.8),13H MICRO G/G**2,	00004250
X/3X,8H BIAS	,3(3X,G16.8),8H MICRO G,	00004260
X/3X,8H BIASV	,3(3X,G16.8),13H (MICRO G)**2,	00004270
X/3X,8H SFP0	,3(3X,G16.8),4H PPM,	00004280
X/3X,8H SFM0	,3(3X,G16.8),4H PPM,	00004290
X/3X,8H SFP1	,3(3X,G16.8),7H PPM /G,	00004300
X/3X,8H SFM1	,3(3X,G16.8),7H PPM /G,	00004310
X/3X,8H RX	,3(3X,G16.8),4H FT ,	00004320
X/3X,8H RY	,3(3X,G16.8),4H FT ,	00004330
X/3X,8H RZ	,3(3X,G16.8),4H FT ,	00004340
X/3X,8H ORDER	,15,/))	00004350
1100 FORMAT(1H+,8X,G16.8)		00004360
1200 FORMAT(6X,25H ** ACC ** DV(FT/SEC)	,3G16.8,/))	00004370
END		00004380

C		00000010
C	03/10/78 DATE OF CURRENT MODULE	00000020
C		00000030
C	THE GYRO MODULE SIMULATES A SINGLE DEGREE OF FREEDOM (SDF) RATE	00000040
C	INTEGRATING GYRO.	00000050
C		00000060
	SUBROUTINE GYROS (T,IENDF,WBB,WDOT,ABB,	00000070
	X DTHETA)	00000080
C		00000090
	REAL ABB(3)	00000100
	REAL ABG(3)	00000110
	REAL AQUANT	00000120
	REAL A2D	00000130
	REAL BIA	00000140
	REAL BIAS(3)	00000150
	REAL BIASA(3)	00000160
	REAL BIASV(3)	00000170
	REAL BIASTC	00000180
	REAL BUF(17)	00000190
	REAL CO	00000200
	REAL DATA(97)	00000210
	REAL DELI	00000220
	REAL DEN	00000230
	REAL DT	00000240
	REAL DTHETA(3)	00000250
	REAL DTHETI(3)	00000260
	REAL DTHETR(3)	00000270
	REAL DTI	00000280
	REAL EX	00000290
	REAL G	00000300
	REAL GAUSS	00000310
	REAL GBIAS(3)	00000320
	REAL H	00000330
	REAL HT	00000340
	REAL I	00000350
	REAL K	00000360
	REAL KI(3)	00000370
	REAL KO(3)	00000380
	REAL KS(3)	00000390
	REAL KII(3)	00000400
	REAL KSS(3)	00000410
	REAL KIO(3)	00000420
	REAL KOS(3)	00000430
	REAL KIS(3)	00000440
	REAL LAT	00000450
	REAL M1	00000460
	REAL M2	00000470
	REAL MODPOT	00000480
	REAL PBUF(16)	00000490
	REAL PDATA(20)	00000500
	REAL PI	00000510
	REAL QGBX(9)	00000520
	REAL QGBY(9)	00000530
	REAL QGBZ(9)	00000540
	REAL QUANT	00000550
	REAL RE	00000560
	REAL SFM0(3)	00000570
	REAL SFM1(3)	00000580
	REAL SFP0(3)	00000590
	REAL SFP1(3)	00000600

REAL SM0(3)	00000610
REAL SM1(3)	00000620
REAL SP0(3)	00000630
REAL SP1(3)	00000640
REAL SPARE1	00000650
REAL SPARE2	00000660
REAL T	00000670
REAL TOTDET	00000680
REAL TGYR	00000690
REAL THDOT(3)	00000700
REAL THN	00000710
REAL THETA(3)	00000720
REAL TPRNT	00000730
REAL TPRMOD	00000740
REAL TRANS1(3)	00000750
REAL TRANTC(3)	00000760
REAL TT	00000770
REAL TTDET(3)	00000780
REAL T1	00000790
REAL WBB(3)	00000800
REAL WBG(3)	00000810
REAL WBOOT(3)	00000820
REAL WDOT(3)	00000830
REAL WE	00000840
C	00000850
INTEGER IENDF	00000860
INTEGER IFILE	00000870
INTEGER INITSW	00000880
C	00000890
INTEGER OFILE	00000900
INTEGER OUTSW	00000910
INTEGER PFILE	00000920
C	00000930
EQUIVALENCE (DATA(1), DT)	00000940
EQUIVALENCE (DATA(2), PRNTSW)	00000950
EQUIVALENCE (DATA(3), OUTSW)	00000960
EQUIVALENCE (DATA(4), XFILE)	00000970
EQUIVALENCE (DATA(5), SPARE1)	00000980
EQUIVALENCE (DATA(6), SPARE2)	00000990
EQUIVALENCE (DATA(7), CO)	00001000
EQUIVALENCE (DATA(13), QGBX(1))	00001010
EQUIVALENCE (DATA(22), QGBY(1))	00001020
EQUIVALENCE (DATA(31), QGBZ(1))	00001030
EQUIVALENCE (DATA(40), H)	00001040
EQUIVALENCE (DATA(42), QUANT)	00001050
EQUIVALENCE (DATA(43), BIAS(1))	00001060
EQUIVALENCE (DATA(46), K)	00001070
EQUIVALENCE (DATA(47), I)	00001080
EQUIVALENCE (DATA(48), DELI)	00001090
EQUIVALENCE (DATA(50), KI(1))	00001100
EQUIVALENCE (DATA(53), KO(1))	00001110
EQUIVALENCE (DATA(56), KS(1))	00001120
EQUIVALENCE (DATA(59), KII(1))	00001130
EQUIVALENCE (DATA(62), KSS(1))	00001140
EQUIVALENCE (DATA(65), KIO(1))	00001150
EQUIVALENCE (DATA(68), KIS(1))	00001160
EQUIVALENCE (DATA(71), KOS(1))	00001170
EQUIVALENCE (DATA(74), BIASV(1))	00001180
EQUIVALENCE (DATA(77), SFP0(1))	00001190
EQUIVALENCE (DATA(80), SFP0(1))	00001200

EQUIVALENCE (DATA(83),MODPDT)	00001210
EQUIVALENCE (DATA(84),ORDER)	00001220
EQUIVALENCE (DATA(85),TRANSI(1))	00001230
EQUIVALENCE (DATA(88),TRANTC(1))	00001240
EQUIVALENCE (DATA(91),SFP1(1))	00001250
EQUIVALENCE (DATA(94),SPM1(1))	00001260
EQUIVALENCE (DATA(97),BIASTC)	00001270
C	00001280
EQUIVALENCE (PDATA(1), WE)	00001290
EQUIVALENCE (PDATA(2), RE)	00001300
EQUIVALENCE (PDATA(3), G)	00001310
EQUIVALENCE (PDATA(4), PRNTDT)	00001320
EQUIVALENCE (PDATA(5), PBUF(1))	00001330
C	00001340
DATA AM /0./	00001350
DATA GBIAS /3*0./	00001360
DATA PI /3.1415926535897 /	00001370
DATA THDOT/3*0./	00001380
DATA THETA/3*0./	00001390
DATA TPRNT/0.0/	00001400
DATA TPRMOD/0.0/	00001410
DATA TTTHET/3*0./	00001420
DATA IFILE/40/	00001430
DATA INITSW/0/	00001440
DATA ITHET /0/	00001450
DATA IX /5/	00001460
DATA K10 /0/	00001470
DATA K11 /0/	00001480
DATA K12 /0/	00001490
DATA OFILE/6/	00001500
DATA PFILE/7/	00001510
C	00001520
IF (IENDF.EQ.1) RETURN	00001530
IF (INITSW.EQ.0) GO TO 500	00001540
IF (T.LT.TGYR-.0005) RETURN	00001550
C	00001560
C	00001570
DO 400 I1=1,3	00001580
IF (I1.GT.1) GOTO 110	00001590
C	00001600
C TRANSFORM FROM BODY TO GYRO COORDINATES(FROM XYZ TO IOS)	00001610
C TRANSFORM X GYRO	00001620
C	00001630
CALL MXV(QGBX,ABB,ABG)	00001640
CALL MXV(QGBX,WBB,WBG)	00001650
CALL MXV(QGBX,WDOT,WBDOT)	00001660
GOTO 130	00001670
C	00001680
110 IF(I1.GT.2) GOTO 120	00001690
C	00001700
C TRANSFORM Y GYRO	00001710
C	00001720
CALL MXV(QGBY,ABD,ABG)	00001730
CALL MXV(QGBY,WDB,WBG)	00001740
CALL MXV(QGBY,WDOT,WBDOT)	00001750
GOTO 130	00001760
C	00001770
C TRANSFORM Z GYRO	00001780
C	00001790
120 CALL MXV(QGBZ,ABB,ABG)	00001800

CALL MXV(QGBZ,WBB,WBG)	00001810
CALL MXV(QGBZ,WDOT,WSDOT)	00001820
C	00001830
130 CONTINUE	00001840
C	00001850
C IF THIS IS THE FIRST PASS, INITIALIZE THETA	00001860
C	00001870
IF (K10.NE.0) GOTO 135	00001880
WSDOT(2)=0.	00001890
THETA(I1)=WBG(1)*H/K	00001900
135 CONTINUE	00001910
C	00001920
C IF THE BIAS VARIANCE IS ZERO, BRANCH AROUND RANDOM NUMBER GENERATOR	00001930
C	00001940
IF (K11.EQ.0) GOTO 138	00001950
BI=BIASA(I1)	00001960
C	00001970
C COMPUTE EXPONENTIALLY CORRELATED RANDOM BIAS	00001980
C	00001990
GBIAS(I1)=GBIAS(I1)*EX+GAUSS(AM,BI)	00002000
BIA=BIAS(I1)+GBIAS(I1)	00002010
GOTO 139	00002020
C	00002030
138 BIA=BIAS(I1)	00002040
C	00002050
139 CONTINUE	00002060
C	00002070
C CALCULATE THETA	00002080
C	00002090
DEN=K*DELI*(WBG(3)*WBG(3)-WBG(1)*WBG(1))+WBG(3)*H	00002100
M1= H*(WBG(1)-BIA-TRANSL(I1)*EXP(-T/TRANTC(I1)))+DELI*WBG(3)*WBG	00002110
X (1)-I*WSDOT(2)	00002120
C	00002130
C IF THE ACCELERATION AND ACCELERATION**2 TERMS ARE 0, BRANCH AROUND	00002140
C	00002150
M2=0.0	00002160
IF (K12.NE.1) GOTO 850	00002170
M2=H*(-KI(I1)*ABG(1)-KS(I1)*ABG(3)-KO(I1)*	00002180
X ABG(2)-KSS(I1)*ABG(3)*ABG(3)-KII(I1)*ABG(1)*ABG(1)-KIS(I1)*ABG	00002190
X (1)*ABG(3)-KIO(I1)*ABG(1)*ABG(2)-KOS(I1)*ABG(2)*ABG(3))	00002200
850 CONTINUE	00002210
C	00002220
IF (NORDER.NE.0) GOTO 700	00002230
C	00002240
C CALCULATE THETA USING A PERFORMANCE MODEL	00002250
C	00002260
THN=(M1+M2)/DEN	00002270
C	00002280
GOTO 800	00002290
C	00002300
700 IF (NORDER.NE.1) GOTO 750	00002310
C	00002320
C CALCULATE THETA USING A FIRST ORDER DIFFERENTIAL EQUATION	00002330
C	00002340
THN=THETA(I1)+D/T*(-DEN*THETA(I1)+M1+M2)/CO	00002350
C	00002360
GOTO 800	00002370
C	00002380
750 IF (NORDER.NE.2) GOTO 2000	00002390
C	00002400

C CALCULATE THETA USING A SECOND ORDER DIFFERENTIAL EQUATION	00002410
C	00002420
THN=THETA(I1)+DT*THDOT(I1)	00002430
THDOT(I1)=THDOT(I1)+(-CO*THDOT(I1)-DEN*THETA(I1)+M1+M2)*DTI	00002440
C	00002450
800 CONTINUE	00002460
C	00002470
C CALCULATE QUANTIZED DELTA THETA	00002480
C	00002490
TTHET(I1)=THN+TTHET(I1)	00002500
IF (TTHET(I1).LT.0.) GOTO 140	00002510
T1=1.+SP0(I1)+SP1(I1)*THN/H*K	00002520
GOTO 145	00002530
C	00002540
140 T1=1.+SM0(I1)+SM1(I1)*THN/H*K	00002550
C	00002560
145 ITHET=TTHET(I1)/AQUANT/T1	00002570
TT=ITHET	00002580
TTHET(I1)=TTHET(I1)-TT*T1*AQUANT	00002590
THETA(I1)=THN	00002600
C	00002610
C INTEGRATE DTHETA	00002620
C	00002630
C	00002640
DTHETA(I1)=DTHETA(I1)+TT*AQUANT*DT/H*K	00002650
C	00002660
151 PRINT I51,THN,THDOT(I1),ITHET,TTHET(I1),ABG(1),WBG(1),THETA(I1)	00002670
400 CONTINUE	00002680
C	00002690
C PRINT OUTPUT AND CONTROL	00002700
C	00002710
IF (PRNTDT.GT.0) GO TO 960	00002720
IF (MODPDT.EQ.0) GO TO 999	00002730
C	00002740
IF (T.LT.TPRMOD-.0005) GO TO 999	00002750
TPRMOD=TPRMOD+MODPDT	00002760
GO TO 970	00002770
C	00002780
960 CONTINUE	00002790
IF (T.LT.TPRNT-.0005) GO TO 999	00002800
TPRNT=TPRNT+PRNTDT	00002810
C	00002820
970 CONTINUE	00002830
IF (PRNTSW.LT.1.) GO TO 999	00002840
C	00002850
WRITE(OFIL,1200) DTHETA	00002860
C	00002870
C	00002880
C	00002890
999 CONTINUE	00002900
K10=1	00002910
TGYR=T+DT	00002920
RETURN	00002930
C	00002940
C	00002950
2000 WRITE (OFIL,2010)	00002960
2010 FORMAT ("ORDER NOT PROPERLY SPECIFIED")	00002970
STOP	00002980
C	00002990
C	00003000

C READ THE DATA FILES AND INITILIZE THE DATA	00003010
C	00003020
500 CONTINUE	00003030
REWIND IFILE	00003040
REWIND PFILE	00003050
501 READ (IFILE,1000) IQ,DATA(IQ)	00003060
IF (EOF(IFILE)) 502,501	00003070
502 CONTINUE	00003080
503 READ (PFILE,1000) IQ,PDATA(IQ)	00003090
IF (EOF(PFILE)) 510,503	00003100
510 CONTINUE	00003110
C	00003120
REWIND IFILE	00003130
REWIND PFILE	00003140
OFILE=XFILE	00003150
C	00003160
NORDER=ORDER	00003170
C	00003180
WRITE (OFILE,1010) DT,PRNTSW,MODPDT,PRNTDT,OUTSW,OFILE,	00003190
X (QGBX(I2),I2=1,9),	00003200
X (QGBY(I2),I2=1,9),(QGBZ(I2),I2=1,9)	00003210
WRITE(OFILE,1011) QUANT	00003220
WRITE (OFILE,1012) K,I,DELI,CO,H,KI,KO,KS,KII	00003230
X ,KSS,KIO,KIS ,KOS,BIAS,BIASV,SFP0,SFM0,SFP1	00003240
WRITE (OFILE,1013) SFM1,TRANS1,TRANTC,BIASTC,NORDER	00003250
C	00003260
DO 519 J1=1,3	00003270
SFP0(J1)=SFP0(J1)*1.E-6	00003280
SFP1(J1)=SFP1(J1)*1.E-6	00003290
SFM1(J1)=SFM1(J1)*1.E-6	00003300
SFM0(J1)=SFM0(J1)*1.E-6	00003310
519 CONTINUE	00003320
C	00003330
AQUANT=QUANT*PI/6.48E5*H/DT/K	00003340
EX=EXP(-DT/BIASTC)	00003350
C	00003360
DO 529 I3=1,3	00003370
OTHEA(I3)=0.0	00003380
TRANS1(I3)=TRANS1(I3)*4.85E-6	00003390
BIASA(I3)=SQRT(BIASV(I3)*(1-EX*EX))*4.85E-6	00003400
BIAS(I3)=BIAS(I3)*4.85E-6	00003410
IF (BIASV(I3).NE.0.) K11=1	00003420
IF(KI(I3).NE.0.) K12=1	00003430
KI(I3)=KI(I3)*4.85E-6/G	00003440
IF(KO(I3).NE.0.) K12=1	00003450
KO(I3)=KO(I3)*4.85E-6/G	00003460
IF(KS(I3).NE.0.) K12=1	00003470
KS(I3)=KS(I3)*4.85E-6/G	00003480
IF(KII(I3).NE.0.) K12=1	00003490
KII(I3)=KII(I3)*4.85E-6/G/G	00003500
IF(KSS(I3).NE.0.) K12=1	00003510
KSS(I3)=KSS(I3)*4.85E-6/G/G	00003520
IF(KIO(I3).NE.0.) K12=1	00003530
KIO(I3)=KIO(I3)*4.85E-6/G/G	00003540
IF(KIS(I3).NE.0.) K12=1	00003550
KIS(I3)=KIS(I3)*4.85E-6/G/G	00003560
IF(KOS(I3).NE.0.) K12=1	00003570
KOS(I3)=KOS(I3)*4.85E-6/G/G	00003580
529 CONTINUE	00003590
C	00003600

CO=CO*2.37E-6	00003610
DELI=DELI*2.37E-6	00003620
K=K*2.37E-6	00003630
I=I*2.37E-6	00003640
DTI=DT/I	00003650
H=H*2.37E-6	00003660
C	00003670
INITSW=1	00003680
TCYR=T+DT	00003690
RETURN	00003700
C	00003710
C	00003720
1000 FORMAT (I5,1X,F20.10)	00003730
1010 FORMAT(30H GYROSCOPE INITIALIZATION ,	00003740
X/3X,8H DT ,3X,G16.8,3X,4H SEC,	00003750
X/3X,8H PRNTSW ,3X,G16.8,	00003760
X/3X,8H MODPDT ,3X,G16.8,	00003770
X/3X,8H PRNTDT ,3X,G16.8,	00003780
X/3X,8H OUTSW ,3X,G16.8,	00003790
X/3X,8H OFILE ,3X,I15,/,	00003800
X/3X,8H QGBX = ,3(3X,G16.8),	00003810
X/10X,3(3X,G16.8),	00003820
X/10X,3(3X,G16.8),/,	00003830
X/3X,8H QGBY = ,3(3X,G16.8),	00003840
X/10X,3(3X,G16.8),	00003850
X/10X,3(3X,G16.8),/,	00003860
X/3X,8H QGBZ = ,3(3X,G16.8),	00003870
X/10X,3(3X,G16.8),	00003880
X/10X,3(3X,G16.8),/,	00003890
1011 FORMAT(00003900
X/3X,8H QUANT ,3X,G16.8,7H ARCSEC,/,	00003910
1012 FORMAT (3X,8H K ,3X,G16.8,16H GM CM**2/SEC**2,	00003920
X/3X,8H I , 3X,G16.8,9H GM CM**2,	00003930
X/3X,8H DELI ,3X,G16.8,9H GM CM**2,	00003940
X/3X,8H CO ,3X,G16.8,16H DYNE CM/RAD/SEC,	00003950
X/3X,8H H ,3X,G16.8,13H GM CM**2/SEC,/,	00003960
X/21X,"X",18X,"Y",18X,"Z",	00003970
X/3X,8H KI ,3(3X,G16.8),9H DEG/HR/G,	00003980
X/3X,8H KO ,3(3X,G16.8),9H DEG/HR/G,	00003990
X/3X,8H KS ,3(3X,G16.8),9H DEG/HR/G,	00004000
X/3X,8H KII ,3(3X,G16.8),16H DEG/HR/G**2,	00004010
X/3X,8H KSS ,3(3X,G16.8),16H DEG/HR/G**2,	00004020
X/3X,8H KIO ,3(3X,G16.8),16H DEG/HR/G**2,	00004030
X/3X,8H KIS ,3(3X,G16.8),16H DEG/HR/G**2,	00004040
X/3X,8H KOS ,3(3X,G16.8),16H DEG/HR/G**2,	00004050
X/3X,8H BIAS ,3(3X,G16.8),7H DEG/HR,	00004060
X/3X,8H BIASV ,3(3X,G16.8),12H (DEG/HR)**2,	00004070
X/3X,8H SFP0 ,3(3X,G16.8),4H PPM,	00004080
X/3X,8H SFM0 ,3(3X,G16.8),4H PPM,	00004090
X/3X,8H SFP1 ,3(3X,G16.8),13H PPM/RAD/SEC)	00004100
1013 FORMAT(3X,8H SFM1 ,3(3X,G16.8),13H PPM/RAD/SEC ,/,	00004110
X/3X,8H TRANS1 ,3(3X,G16.8),7H DEG/HR,	00004120
X/3X,8H TRANTC ,3(3X,G16.8),4H SEC,/,	00004130
X/3X,8H BIASTC , 3X,G16.8 ,4H SEC,	00004140
X/3X,8H ORDER ,3X,I7,/,	00004150
1100 FORMAT(1H+,8X,G16.8)	00004160
1200 FORMAT(6X,25H ** GYR ** DTHETA(RADS) ,3G16.8,/,	00004170
END	00004180

```

C                                     00000010
C 3/15/78 DATE OF CURRENT MODULE    00000020
C                                     00000030
C 10/18/77 CHANGE FORMATS TO G16.8  00000040
C                                     00000050
C*****   *** THIS MODULE SIMULATES THREE RING LASER GYROS ***   *****00000060
C                                     00000070
C 02/03/78 DATE OF CURRENT MODULE.  DGR 00000080
C                                     00000090
C      SUBROUTINE GYROS ( T,IENDF,WBB,WDOT,ABB,
C                        OTHETA )      00000100
C                                     00000110
C                                     00000120
C      .WBB      - ANGULAR RATE OF BODY IN BODY COORDINATES. 00000130
C      .DTHETA   - INCREMENTAL ROTATION GENERATED BY GYROS. 00000140
C      .WDOT,ABB - NOT USED.          00000150
C                                     00000160
C      IMPLICIT REAL ( A-Z )          00000170
C      REAL GAUSS                     00000180
C      INTEGER I,IENDF,INITSW,IFILE,IQ,J,K,OFILE,NP,PFILE 00000190
C      DIMENSION ABB(3),ANG(3),ANGRW(3),ANGWN(3),DATA(34),DTHETA(3),
C      . DB(3),DTA(3),DTC(3),EXPD(3),EXPSF(3),KW(9),SF(3),SFTA(3),
C      . SFTC(3),WBB(3),W(3),WDOT(3),PBUF(16),PDATA(20) 00000200
C                                     00000210
C                                     00000220
C      EQUIVALENCE (DATA(1), DT)      00000230
C      EQUIVALENCE (DATA(2), PRNTSW)  00000240
C      EQUIVALENCE (DATA(3), OUTSW)   00000250
C      EQUIVALENCE (DATA(4), XFILE)   00000260
C      EQUIVALENCE (DATA(5), SPARE1)  00000270
C      EQUIVALENCE (DATA(6), SPARE2)  00000280
C      EQUIVALENCE (DATA(7), MODPDT)  00000290
C      EQUIVALENCE (DATA(8), DB(1))   00000300
C      EQUIVALENCE (DATA(11), DTA(1)) 00000310
C      EQUIVALENCE (DATA(14), DTC(1)) 00000320
C      EQUIVALENCE (DATA(17), SFTA(1)) 00000330
C      EQUIVALENCE (DATA(20), SFTC(1)) 00000340
C      EQUIVALENCE (DATA(23), KW(1))  00000350
C      EQUIVALENCE (DATA(32), STDWN)  00000360
C      EQUIVALENCE (DATA(33), STCRW)  00000370
C      EQUIVALENCE (DATA(34), Q)      00000380
C                                     00000390
C      EQUIVALENCE (PDATA(1), WE)     00000400
C      EQUIVALENCE (PDATA(2), RE)     00000410
C      EQUIVALENCE (PDATA(3), G)      00000420
C      EQUIVALENCE (PDATA(4), PRNTDT) 00000430
C      EQUIVALENCE (PDATA(5), PBUF(1)) 00000440
C                                     00000450
C      DATA DTR /.01745329251994330/ 00000460
C      DATA INITSW /0/               00000470
C      DATA IFILE /40/               00000480
C      DATA OFILE /6/               00000490
C      DATA ANG /3*0./              00000500
C      DATA ANGRW /3*0./            00000510
C      DATA PFILE /7/               00000520
C      DATA TPRNT /0.0/             00000530
C      DATA TPRMOD /0.0/            00000540
C      IF (IENDF.EQ.1) RETURN          00000550
C      IF (INITSW.EQ.0) GO TO 500      00000560
C      IF (T.LT.TGYR-.0001) RETURN    00000570
C                                     00000580
C                                     00000590
C                                     00000600

```

```

C-----BEGIN NORMAL COMPUTATIONS-----00000610
C**TRANSIENT DRIFT AND TRANSIENT SCALE FACTOR: DTA,SFTA.00000620
100 DO 105 I = 1,300000630
    IF ( ABS( DTA(I) ) .LT. 1.E-10 ) DTA(I) = 0.00000640
    IF ( ABS( SFTA(I) ) .LT. 1.E-10 ) SFTA(I) = 0.00000650
    DTA(I) = DTA(I)*EXPDI(I)00000660
    SFTA(I) = SFTA(I)*EXPSF(I)00000670
    K=4*(I-1)+100000680
    KW(K) = SF(I) + SFTA(I)00000690
105 CONTINUE00000700
C**COMPUTE WHITE AND RANDOM WALK ANGLE NOISE: ANGWN,ANGRW.00000710
    DO 110 I = 1,300000720
    ANGWN(I) = STDWN*GAUSS(0.0,1.0)00000730
110 ANGRW(I) = KRW*GAUSS(0.0,1.0)00000740
C**COMPUTE THE CONTINUOUS ANGLE: ANG.00000750
    CALL MXV ( KW,WBB,W )00000760
    DO 112 I = 1,300000770
    W(I) = WBB(I) + W(I) + DB(I) + DTA(I)00000780
    ANG(I) = ANG(I) + W(I)*DT00000790
112 ANG(I) = ANG(I) + ANGWN(I) + ANGRW(I)00000800
C**COMPUTE QUANTIZED INDICATED ROTATION: DTHETA.00000810
C    NP = NUMBER OF OUTPUT PULSES.00000820
C    Q = ANGULAR QUANTIZATION.00000830
C    NOTE: THE NAV ROUTINES SETS (DTHETA) TO ZERO EACH TIME THE00000840
C    PLATFORM ATTITUDE MATRIX IS COMPUTED.00000850
    DO 115 I = 1,300000860
    IF ( Q .GT. 0. ) GO TO 11300000870
    QANG = ANG(I)00000880
    GO TO 11400000890
113 NP = ANG(I)/Q00000900
    QANG = NP*Q00000910
114 ANG(I) = ANG(I) - QANG - ANGWN(I)00000920
115 DTHETA(I) = DTHETA(I) + QANG00000930
C00000940
C    OUTPUT AND PRINT CONTROL00000950
C00000960
    IF ( PRNTDT.GT.0 ) GO TO 9600000970
    IF ( MODPDT.EQ.0 ) GO TO 9990000980
C00000990
    IF ( T.LT.TPRMOD-.0005 ) GO TO 9990001000
    TPRMOD=TPRMOD+MODPDT00010010
    GO TO 9700001020
C00001030
960 CONTINUE00001040
    IF ( T.LT.TPRNT-.0005 ) GO TO 9990001050
    TPRNT=TPRNT+PRNTDT00001060
C00001070
970 CONTINUE00001080
    IF ( FRNTSW.LT.1 ) GO TO 9990001090
C00001100
    WRITE(OFIL,1200) DTHETA00001110
C00001120
999 CONTINUE00001130
    TGYR=T+DT00001140
    RETURN00001150
C00001160
C-----INITIALIZATION SECTION-----00001170
C**READ THE INPUT DATA FILE: IFILE.00001180
C    DT - SIMULATION TIME STEP ( SEC )00001190
C    DB - BIAS DRIFT (DEG/HR)00001200

```

C	.DTA	- DRIFT TRANSIENT AMPLITUDE	(DEG/HR)	00001210
C	.DTC	- DRIFT TRANSIENT TIME CONSTANT	(MIN)	00001220
C	.SFTA	- SCALE FACTOR TRANSIENT AMPLITUDE	(PPM)	00001230
C	.SFTC	- SCALE FACTOR TRANSIENT TIME CONSTANT	(MIN)	00001240
C	.KW	- SF AND GYRO IA MISALIGNMENT MATRIX	(PPM)	00001250
C	.STDWN	- STD WHITE ANGLE NOISE	(SEC)	00001260
C	.STDRW	- RANDOM WALK ANGLE MAGNITUDE	DEG/SQRT(HR)	00001270
C	.Q	- ANGULAR QUANTIZATION	(SEC)	00001280
C				00001290
C				00001300
	500	CONTINUE		00001310
		REWIND IFILE		00001320
		REWIND PFILE		00001330
	501	READ (IFILE,1000) IQ,DATA(IQ)		00001340
		IF (EOF(IFILE)) 502,501		00001350
	502	CONTINUE		00001360
	503	READ (PFILE,1000) IQ,PDATA(IQ)		00001370
		IF (EOF(PFILE)) 510,503		00001380
	510	CONTINUE		00001390
C				00001400
		REWIND IFILE		00001410
		REWIND PFILE		00001420
		OFILE=XFILE		00001430
C				00001440
		WRITE (OFILE,1010) DT,PRNTSW,PRNTDT,OUTSW,OFILE,MODPOT		00001450
		WRITE(OFILE,1011) Q,STDWN,STDRW		00001460
		WRITE (OFILE,1012) DB,DTA,DTC,SFTA,SFTC,KW		00001470
C				00001480
	C**SCALE THE INPUT DATA TO INTERNAL PROGRAM UNITS.			00001490
	0	DO 10 I = 1,3		00001500
		DB(I) = DB(I)*DTR/3600.		00001510
		DTA(I) = DTA(I)*DTR/3600.		00001520
		DTC(I) = DTC(I)*60.		00001530
		SFTA(I) = SFTA(I)*1.E-06		00001540
		SFTC(I) = SFTC(I)*60.		00001550
		DO 10 J = 1,3		00001560
		K=3*(I-1)+J		00001570
		KW(K) = KW(K)*1.E-05		00001580
	10	CONTINUE		00001590
		STDWN = STDWN*DTR/3600.		00001600
		KRW = (STDRW*DTR)*SQRT(DT/3600.)		00001610
		Q = Q*DTR/3600.		00001620
C				00001630
		DO 11 I = 1,3		00001640
		K=4*(I-1)+1		00001650
		SF(I) = KW(K)		00001660
		EXPDI(I) = 0.		00001670
		EXPSF(I) = 0.		00001680
		IF (DTC(I) .GT. 0.) EXPDI(I) = EXP(-DT/DTC(I))		00001690
		IF (SFTC(I) .GT. 0.) EXPSF(I) = EXP(-DT/SFTC(I))		00001700
	11	CONTINUE		00001710
C				00001720
		INITSW=1		00001730
		TSYR=Y+DT		00001740
		RETURN		00001750
	C**END INITIALIZATION.			00001760
C				00001770
C				00001780
C				00001790
	1000	FORMAT (I5,F20.10)		00001800

1010	FORMAT(30H LASER GYRO INITIALIZATION	00001810
	X/3X,8H DT ,3X,G16.8,3X,4H SEC,	00001820
	X/3X,8H PRNTSW ,3X,G16.8,	00001830
	X/3X,8H PRNTDT ,3X,G16.8,	00001840
	X/3X,8H OUTSW ,3X,G16.8,	00001850
	X/3X,8H OFILE ,3X,I7,	00001860
	X/3X,8H MODPDT ,3X,G16.8,3X,4H SEC,//)	00001870
1011	FORMAT(00001880
	X/3X,8H Q ,3X,G16.8,4H SEC,	00001890
	X/3X,8H STDWN ,3X,G16.8,4H SEC,	00001900
	X/3X,8H STDRW ,3X,G16.8,13H DEG/SQRT(HR),//)	00001910
1012	FORMAT (00001920
	X/21X,"X",18X,"Y",18X,"Z",	00001930
	X/3X,8H DB ,3(3X,G16.8),7H DEG/HR,	00001940
	X/3X,8H DTA ,3(3X,G16.8),7H DEG/HR,	00001950
	X/3X,8H DTC ,3(3X,G16.8),4H MIN,	00001960
	X/3X,8H SFTA ,3(3X,G16.8),4H PPM,	00001970
	X/3X,8H SFTC ,3(3X,G16.8),4H MIN,	00001980
	X//3X,8H KW= ,3(3X,G16.8),4H PPM,	00001990
	X/11X, 3(3X,G16.8),	00002000
	X/11X, 3(3X,G16.8),//)	00002010
1200	FORMAT(6X,25H ** GYR ** DTHETA(RADS) ,3(3X,G20.12))	00002020
	END	00002030

C		00000010
C	03/10/78 DATE OF CURRENT MODULE	00000020
C		00000030
C	THE ALTIMETER MODULE SIMULATES A BAROMETRIC ALTIMETER.	00000040
C		00000050
	SUBROUTINE ALTI (T,IENDF,ALT,V,	00000060
	X ALTO)	00000070
C		00000080
	REAL ALT	00000090
	REAL ALTA2	00000100
	REAL ALTO	00000110
	REAL AM	00000120
	REAL DATA(12)	00000130
	REAL DT	00000140
	REAL EX	00000150
	REAL ECX1	00000160
	REAL GAUSS	00000170
	REAL MODPOT	00000180
	REAL PBUF(16)	00000190
	REAL PDATA(20)	00000200
	REAL PRNTDT	00000210
	REAL TPRMOD	00000220
	REAL TPRNT	00000230
	REAL V(3)	00000240
	REAL VI2	00000250
	REAL X	00000260
C		00000270
	INTEGER IENDF	00000280
	INTEGER INITSW	00000290
	INTEGER OUTSW	00000300
	INTEGER IFILE	00000310
	INTEGER PFILE	00000320
	INTEGER OFILE	00000330
C		00000340
	EQUIVALENCE (DATA(1), DT)	00000350
	EQUIVALENCE (DATA(2), PRNTSW)	00000360
	EQUIVALENCE (DATA(3), OUTSW)	00000370
	EQUIVALENCE (DATA(4), XFILE)	00000380
	EQUIVALENCE (DATA(5), MODPOT)	00000390
	EQUIVALENCE (DATA(6), NOISSW)	00000400
	EQUIVALENCE (DATA(7), TC)	00000410
	EQUIVALENCE (DATA(8), U0)	00000420
	EQUIVALENCE (DATA(9), U1)	00000430
	EQUIVALENCE (DATA(10), U2)	00000440
	EQUIVALENCE (DATA(11), U3)	00000450
	EQUIVALENCE (DATA(12), U4)	00000460
C		00000470
	EQUIVALENCE (PDATA(1), WE)	00000480
	EQUIVALENCE (PDATA(2), RE)	00000490
	EQUIVALENCE (PDATA(3), G)	00000500
	EQUIVALENCE (PDATA(4), PRNTDY)	00000510
	EQUIVALENCE (PDATA(5), PBUF(1))	00000520
C		00000530
	DATA AM/0.0/	00000540
	DATA TPRMOD/0.0/	00000550
	DATA TPRNT/0.0/	00000560
	DATA X/0.0/	00000570
	DATA INITSH/0/	00000580
	DATA IFILE/60/	00000590
	DATA PFILE/7/	00000600

C	IF (IENDF.EQ.1) RETURN	00000610
	IF (INITSW.EQ.0) GO TO 500	00000620
	IF (T.LT.TALT-.0002) RETURN	00000630
C	ALTO=ALT	00000640
C		00000650
C	ADD NOISE INTO ALTITUDE COMPUTATION IF NOISSW=1	00000660
C		00000670
C	IF (NOISSW.LT.1.) GO TO 110	00000680
C		00000690
C		00000700
C	ALTA2=ALT**2	00000710
	VI2=V(1)*V(1)+V(2)*V(2)+V(3)*V(3)	00000720
	SIGN=SQRT((U0*ALTA2*ALTA2+	00000730
X	U1*VI2*VI2+U2)*E2X1)	00000740
	SIGR=SQRT(U3*ALTA2+U4)	00000750
C		00000760
C	CALCULATE EXPONENTIALLY CORRELATED NOISE TERM	00000770
C		00000780
C	X=EX*X+GAUSS(AM,SIGN)	00000790
C		00000800
C	CALCULATE BAROMETRIC ALTITUDE	00000810
C		00000820
C	ALTO=ALT+X+GAUSS(AM,SIGR)	00000830
C		00000840
C	110 CONTINUE	00000850
C		00000860
C	OUTPUT AND PRINT CONTROL	00000870
C		00000880
	IF (PRNTDT.GT.0) GO TO 960	00000890
	IF (MODPDT.EQ.0) GO TO 999	00000900
C		00000910
	IF (T.LT.TPRMOD-.0005) GO TO 999	00000920
	TPRMOD=TPRMOD+MODPDT	00000930
	GO TO 970	00000940
C		00000950
	960 CONTINUE	00000960
	IF (T.LT.TPRNT-.0005) GO TO 999	00000970
	TPRNT=TPRNT+PRNTDT	00000980
C		00000990
	970 CONTINUE	00001000
	IF (PRNTSW.LT.1.) GO TO 999	00001010
	WRITE(OFIL,1200) ALTO	00001020
C		00001030
	999 CONTINUE	00001040
	TALT=T+DT	00001050
	RETURN	00001060
C		00001070
C	ALTIMETER INITIALIZATION	00001080
C		00001090
	500 CONTINUE	00001100
C		00001110
	REWIND IFILE	00001120
	REWIND PFILE	00001130
	501 READ (IFILE,1000) IX,DATA(IX)	00001140
	IF (EOF(IFILE)) 502,501	00001150
	502 CONTINUE	00001160
C		00001170
	503 READ (PFILE,1000) IX,POATA(IX)	00001180
		00001190
		00001200

IF (EOF(PFILE)) 510,503	00001210
510 CONTINUE	00001220
C REWIND PFILE	00001230
REWIND IFILE	00001240
C	00001250
ALTO=ALT	00001260
C	00001270
EX=EXP(-DT/TC)	00001280
E2X1=1-EX**2	00001290
C	00001300
C INITIALIZATION OUTPUT AND PRINT CONTROL	00001310
C	00001320
OFIL=XFILE	00001330
WRITE (OFIL,1010) DT,PRNTSW,OUTSW,OFIL,PRNTDT,MODPDT,NOISSW	00001340
WRITE (OFIL,1012) TC,U0,U1,U2,U3,U4	00001350
C	00001360
INITSW=1	00001370
TALT=T+DT	00001380
RETURN	00001390
C	00001400
C END OF INITIALIZATION	00001410
C	00001420
1000 FORMAT (I5,F20.10)	00001430
1010 FORMAT (30H ALTIMETER INITIALIZATION ,	00001440
X/3X,8H DT ,3X,G16.8,3X,4H SEC,	00001450
X/3X,8H PRNTSW ,3X,G16.8,	00001460
X/3X,8H OUTSW ,3X,G16.8,	00001470
X/3X,8H OFIL ,3X,I15,	00001480
X/3X,8H PRNTDT ,3X,G16.8,	00001490
X/3X,8H MODPDT ,3X,G16.8,	00001500
X/3X,8H NOISSW ,3X,I15,//)	00001510
1012 FORMAT(3X,30H ALTIMETER UNCERTAINTIES ,//,	00001520
X/6X,14H TC ,3X,G16.8,	00001530
X/6X,14H U0(FT-2) ,3X,G16.8,	00001540
X/6X,14H U1(SEC4/FT2) ,3X,G16.8,	00001550
X/6X,14H U2(FT2) ,3X,G16.8,	00001560
X/6X,14H U3 ,3X,G16.8,	00001570
X/6X,14H U4(FT2) ,3X,G16.8,//)	00001580
1200 FORMAT(6X,25H ** ALT ** ALT(FT) ,G16.8,//)	00001590
END	00001600
	00001610

C		00000010
C	03/10/78 DATE OF CURRENT MODULE	00000020
C		00000030
C	THE HARDWARE/SOFTWARE INTERFACE MODULE FUNCTIONS AS A BUFFER BETWEEN	00000040
C	THE INSS SEQUENCE FAST AND SLOW CYCLES.	00000050
C		00000060
	SUBROUTINE RDR (T,IENDF,DTHETA,DV,	00000070
X	DTHETO,DVO)	00000080
C		00000090
	REAL DATA(6)	00000100
	REAL DTHETA(3)	00000110
	REAL DTHETO(3)	00000120
	REAL DV(3)	00000130
	REAL DVO(3)	00000140
	REAL MODPDT	00000150
	REAL SPARE1	00000160
	REAL PBUF(16)	00000170
	REAL PDATA(20)	00000180
	REAL PRNTDT	00000190
	REAL T	00000200
	REAL TPRNT	00000210
	REAL TPRMOD	00000220
	REAL TEMP(3)	00000230
	REAL TRDR	00000240
C		00000250
	INTEGER IENDF	00000260
	INTEGER OFILE	00000270
	INTEGER PFILE	00000280
C		00000290
	EQUIVALENCE (DATA(1), DT)	00000300
	EQUIVALENCE (DATA(2), PRNTSW)	00000310
	EQUIVALENCE (DATA(3), OUTSW)	00000320
	EQUIVALENCE (DATA(4), XFILE)	00000330
	EQUIVALENCE (DATA(5), SPARE1)	00000340
	EQUIVALENCE (DATA(6), MODPDT)	00000350
C		00000360
	EQUIVALENCE (PDATA(1), WE)	00000370
	EQUIVALENCE (PDATA(2), RE)	00000380
	EQUIVALENCE (PDATA(3), G)	00000390
	EQUIVALENCE (PDATA(4), PRNTDT)	00000400
	EQUIVALENCE (PDATA(5), PBUF(1))	00000410
C		00000420
	DATA TPRNT/0.0/	00000430
	DATA TPRMOD/0.0/	00000440
	DATA INITSW/0/	00000450
	DATA IFILE/65/	00000460
	DATA PFILE/7/	00000470
C		00000480
	IF (IENDF.EQ.1) RETURN	00000490
	IF (INITSW.EQ.0) GOTO 500	00000500
	IF (T.LT.TRDR-.0001) RETURN	00000510
C		00000520
C		00000530
	DO 100 I=1,3	00000540
	DVO(I)=DV(I)	00000550
	DV(I)=0.	00000560
	DTHETO(I)=DTHETA(I)	00000570
	DTHETA(I)=0.	00000580
	100 CONTINUE	00000590
C		00000600

C		00000610
C	OUTPUT AND PRINT CONTROL	00000620
C		00000630
	IF (PRNTDT.G7.0) GO TO 960	00000640
	IF (MODPDT.EQ.0) GO TO 999	00000650
C		00000660
	IF (T.LT.TPRMOD-.0005) GO TO 999	00000670
	TPRMOD=TPRMOD+MODPDT	00000680
	GO TO 970	00000690
C		00000700
	960 CONTINUE	00000710
	IF (T.LT.TPRNT-.0005) GO TO 999	00000720
	TPRNT=TPRNT+PRNTDT	00000730
C		00000740
	970 CONTINUE	00000750
	IF (FRNTSW.LT.1) GO TO 999	00000760
	WRITE(OFIL,1200)DVO,DTHETO	00000770
C		00000780
C		00000790
	999 CONTINUE	00000800
	TRDR=TRDR+DT	00000810
	RETURN	00000820
C		00000830
C	READER INITIALIZATION BEGINS HERE	00000840
C		00000850
	500 CONTINUE	00000860
	REWIND IFILE	00000870
	501 READ (IFILE,1000)IX,DATA(IX)	00000880
	IF (EOF(IFILE)) 502,501	00000890
	502 CONTINUE	00000900
	503 READ(PFILE,1000) IX,PDATA(IX)	00000910
	IF (EOF(PFILE)) 510,503	00000920
	510 CONTINUE	00000930
C		00000940
	REWIND IFILE	00000950
	REWIND PFILE	00000960
	OFIL=XFILE	00000970
C		00000980
	WRITE(OFIL,1010)DT,PRNTSW,OUTSW,OFIL,MODPDT,PRNTDT	00000990
C		00010000
	INITSW=1	00001010
	TRDR=T+DT	00001020
	RETURN	00001030
C		00001040
C	END OF INITIALIZATION	00001050
C		00001060
	1000 FORMAT(I5,F20.10)	00001070
	1010 FORMAT(30H READER INITIALIZATION	00001080
	X/3X,8H DT ,3X,G16.8,3X,4H SEC,	00001090
	X/3X,8H PRNTSW ,3X,G16.8,	00001100
	X/3X,8H OUTSW ,3X,G16.8,	00001110
	X/3X,8H OFIL ,3X,I15,	00001120
	X/3X,8H MODPDT ,3X,G16.8,	00001130
	X/3X,8H PRNTDT ,3X,G16.8,/))	00001140
	1200 FORMAT(6X,25H ** RDR ** DV(FY/SEC) ,3G16.8, /	00001150
	X 6X,25H DTHETA(RADS) ,3G16.8,/))	00001160
	END	00001170

C		00000010
C	03/10/78 DATE OF CURRENT MODULE	00000020
C		00000030
C	THE ACCELEROMETER COMPENSATION MODULE COMPENSATES FOR ACCELEROMETER	00000040
C	BIASES AND ERRORS.	00000050
C		00000060
	SUBROUTINE ACOMP (T,IENDF,DVO,DTHETA,	00000070
	X DVA)	00000080
C		00000090
	REAL BIAS(3)	00000100
	REAL CROSS1(3)	00000110
	REAL CROSS2(3)	00000120
	REAL CROSS3(3)	00000130
	REAL DATA(73)	00000140
	REAL DT	00000150
	REAL DOTH(3)	00000160
	REAL DELI	00000170
	REAL DTHETA(3)	00000180
	REAL DTHETP(3)	00000190
	REAL DTHETZ(3)	00000200
	REAL DVA(3)	00000210
	REAL DVO(3)	00000220
	REAL G	00000230
	REAL IXX	00000240
	REAL KII(3)	00000250
	REAL MRC	00000260
	REAL MODPDT	00000270
	REAL PBUF(16)	00000280
	REAL PDATA(20)	00000290
	REAL QABO(9)	00000300
	REAL QABX(9)	00000310
	REAL QABY(9)	00000320
	REAL QABZ(9)	00000330
	REAL QNIS(9)	00000340
	REAL RX(3)	00000350
	REAL RY(3)	00000360
	REAL RZ(3)	00000370
	REAL SFM0(3)	00000380
	REAL SFM1(3)	00000390
	REAL SFP0(3)	00000400
	REAL SFP1(3)	00000410
	REAL SM0(3)	00000420
	REAL SM1(3)	00000430
	REAL SP0(3)	00000440
	REAL SP1(3)	00000450
	REAL SPARE1	00000460
	REAL SPARE2	00000470
	REAL TFRMCD	00000480
	REAL TFRNT	00000490
	REAL WOOT(3)	00000500
	REAL WE	00000510
C		00000520
	INTEGER IENDF	00000530
	INTEGER INITSH	00000540
	INTEGER IFILE	00000550
	INTEGER OFILE	00000560
	INTEGER OUTSH	00000570
	INTEGER PFILE	00000580
C		00000590
	EQUIVALENCE (DATA(1), DT)	00000600

	EQUIVALENCE (DATA(2), PRNTSW)	00000610
	EQUIVALENCE (DATA(3), OUTSW)	00000620
	EQUIVALENCE (DATA(4), XFILE)	00000630
	EQUIVALENCE (DATA(5), SPARE1)	00000640
	EQUIVALENCE (DATA(6), SPARE2)	00000650
	EQUIVALENCE (DATA(7), DELI)	00000660
	EQUIVALENCE (DATA(8), QABX(1))	00000670
	EQUIVALENCE (DATA(17), SFP0(1))	00000680
	EQUIVALENCE (DATA(20), BIAS(1))	00000690
	EQUIVALENCE (DATA(23), QABY(1))	00000700
	EQUIVALENCE (DATA(32), QABZ(1))	00000710
	EQUIVALENCE (DATA(41), MODPDT)	00000720
	EQUIVALENCE (DATA(42), KII(1))	00000730
	EQUIVALENCE (DATA(45), MRC)	00000740
	EQUIVALENCE (DATA(46), QMIS(1))	00000750
	EQUIVALENCE (DATA(55), SFM0(1))	00000760
	EQUIVALENCE (DATA(58), SFP1(1))	00000770
	EQUIVALENCE (DATA(61), SFM1(1))	00000780
	EQUIVALENCE (DATA(64), RX(1))	00000790
	EQUIVALENCE (DATA(67), RY(1))	00000800
	EQUIVALENCE (DATA(70), RZ(1))	00000810
	EQUIVALENCE (DATA(73), IXX)	00000820
C		00000830
	EQUIVALENCE (PDATA(1), WE)	00000840
	EQUIVALENCE (PDATA(2), RE)	00000850
	EQUIVALENCE (PDATA(3), G)	00000860
	EQUIVALENCE (PDATA(4), PRNTDT)	00000870
	EQUIVALENCE (PDATA(5), PBUF(1))	00000880
C		00000890
	DATA CROSS3/3*0./	00000900
	DATA TPRMOD/0.0/	00000910
	DATA TFRNT/0./	00000920
	DATA INITSW/0/	00000930
	DATA IFILE/67/	00000940
	DATA K10/0/	00000950
	DATA PFILE/7/	00000960
C		00000970
	IF (IENDF.EQ.1) RETURN	00000980
	IF (INITSW.EQ.0) GO TO 500	00000990
	IF (T.LT.TCAC-.0001) RETURN	00001000
C		00001010
C		00001020
C	COMPENSATE FOR SCALE FACTOR ERROR AND BIAS	00001030
C		00001040
	IF (K10.EQ.0) GOTO 25	00001050
C		00001060
	DO 20 I3=1,3	00001070
20	DDTH(I3)=DTHETA(I3)-DTHETP(I3)	00001080
	CALL MXV(QAB0,DDTH,WDOT)	00001090
	DO 22 I2=1,3	00001100
	WDOT(I2)=WDOT(I2)/DT	00001110
22	CONTINUE	00001120
C		00001130
25	CONTINUE	00001140
C		00001150
	DO 100 I=1,3	00001160
	IF (DVO(I).LT.0.) GOTO 26	00001170
	Y1=1.+SP0(I)+SP1(I)*DVO(I)/DT	00001180
	GOTO 27	00001190
C		00001200

26	T1=1.+SM0(I)+SM1(I)*DVO(I)/DT	00001210
C		00001220
27	CONTINUE	00001230
	DVO(I)=DVO(I)*T1	00001240
	DVO(I)=DVO(I)+BIAS(I)*DT	00001250
C		00001260
	IF(I.NE.1) GOTO 30	00001270
C		00001280
	CROSS1(1)=DTHETA(2)*RX(3)-DTHETA(3)*RX(2)	00001290
	CROSS1(2)=DTHETA(3)*RX(1)-DTHETA(1)*RX(3)	00001300
	CROSS1(3)=DTHETA(1)*RX(2)-DTHETA(2)*RX(1)	00001310
	CROSS2(1)=(DTHETA(2)*CROSS1(3)-DTHETA(3)*CROSS1(2))/DT	00001320
C		00001330
	CALL MXV(QABX,DTHETA,DTHETZ)	00001340
C		00001350
	IF (K10.EQ.0) GOTO 50	00001360
	CROSS3(1)=(DDTH(2)*RX(3)-DDTH(3)*RX(2))/DT	00001370
	GOTO 50	00001380
C		00001390
30	IF(I.NE.2) GOTO 40	00001400
	CROSS1(1)=DTHETA(2)*RY(3)-DTHETA(3)*RY(2)	00001410
	CROSS1(2)=DTHETA(3)*RY(1)-DTHETA(1)*RY(3)	00001420
	CROSS1(3)=DTHETA(1)*RY(2)-DTHETA(2)*RY(1)	00001430
	CROSS2(2)=(DTHETA(3)*CROSS1(1)-DTHETA(1)*CROSS1(3))/DT	00001440
C		00001450
	CALL MXV(QABY,DTHETA,DTHETZ)	00001460
C		00001470
	IF (K10.EQ.0) GOTO 50	00001480
	CROSS3(2)=(DDTH(3)*RY(1)-DDTH(1)*RY(3))/DT	00001490
	GOTO 50	00001500
C		00001510
40	CALL MXV(QABZ,DTHETA,DTHETZ)	00001520
C		00001530
	CROSS1(1)=DTHETA(2)*RZ(3)-DTHETA(3)*RZ(2)	00001540
	CROSS1(2)=DTHETA(3)*RZ(1)-DTHETA(1)*RZ(3)	00001550
	CROSS1(3)=DTHETA(1)*RZ(2)-DTHETA(2)*RZ(1)	00001560
	CROSS2(3)=(DTHETA(1)*CROSS1(2)-DTHETA(2)*CROSS1(1))/DT	00001570
C		00001580
	IF (K10.EQ.0) GOTO 50	00001590
	CROSS3(3)=(DDTH(1)*RZ(2)-DDTH(2)*RZ(1))/DT	00001600
50	CONTINUE	00001610
	DVO(I)=DVO(I)+K11(I)*DVO(I)+DVO(I)/DT-DELY*DTHETZ(3)*DTHETZ(1)/DT	00001620
	X /HRC	00001630
	DVO(I)=DVO(I)-CROSS3(1)-CROSS2(1)	00001640
C		00001650
	IF(K10.EQ.0) GOTO 100	00001660
	DVO(I)=DVO(I)+IXX/HRC*DDOT(I)	00001670
100	CONTINUE	00001680
	K10=1	00001690
C		00001700
	DO 160 I6=1,3	00001710
	160 DTHETP(I6)=DTHETA(I6)	00001720
C		00001730
C	COMPENSATE FOR X,Y, AND Z MISALIGNMENT	00001740
C		00001750
	CALL MXV(QMIS,DVO,DVA)	00001760
C		00001770
C	OUTPUT AND PRINT CONTROL	00001780
C		00001790
	IF (PRINTOT.GT.0) GO TO 960	00001800

IF (MODPDT.EQ.0) GO TO 999	00001810
C	00001820
IF (T.LT.TPRMOD-.0005) GO TO 999	00001830
TPRMOD=TPRMOD+MODPDT	00001840
GO TO 970	00001850
C	00001860
960 CONTINUE	00001870
IF (T.LT.TPRNT-.0005) GO TO 999	00001880
TPRNT=TPRNT+PRNTDT	00001890
C	00001900
970 CONTINUE	00001910
IF (PRNTSW.LT.1.) GO TO 999	00001920
WRITE(OFIL,1200) DVA	00001930
C	00001940
999 CONTINUE	00001950
TCAC=T+DT	00001960
RETURN	00001970
C	00001980
C ACC COMPENSATION MODULE INITIALIZATION	00001990
C	00002000
500 CONTINUE	00002010
REWIND IFILE	00002020
REWIND PFILE	00002030
501 READ (IFILE,1000) IX,DATA(IX)	00002040
IF (EOF(IFILE)) 502,501	00002050
502 CONTINUE	00002060
503 READ (PFILE,1000) IX,PDATA(IX)	00002070
IF (EOF(PFILE)) 504,503	00002080
504 CONTINUE	00002090
C	00002100
REWIND IFILE	00002110
REWIND PFILE	00002120
OFIL=XFILE	00002130
C	00002140
WRITE (OFIL,1010) DT,PRNTSW,OUTSW,OFIL,SPARE1,MODPDT,PRNTDT,DEL	00002150
X,BIAS,KII,MRC,IXX,QABX,QABY,QABZ	00002160
WRITE (OFIL,1011) QMIS,SFMO,SFMI,SFPO,SFPI,RX,RY,RZ	00002170
C	00002180
C	00002190
DO 600 I2=1,3	00002200
BIAS(I2)=BIAS(I2)*1.E-6*G	00002210
KII(I2)=KII(I2)*1.E-6/G	00002220
600 CONTINUE	00002230
C	00002240
QABO(1)=QABX(4)	00002250
QABO(2)=QABX(5)	00002260
QABO(3)=QABX(6)	00002270
QABO(4)=QABY(4)	00002280
QABO(5)=QABY(5)	00002290
QABO(6)=QABY(6)	00002300
QABO(7)=QABZ(4)	00002310
QABO(8)=QABZ(5)	00002320
QABO(9)=QABZ(6)	00002330
IXX=IXX*2.37E-6	00002340
DELI=DELI*2.37E-6	00002350
MRC=MRC*7.23E-5	00002360
DO 650 I1=1,3	00002370
SP0(I1)=SFPO(I1)*1.E-6	00002380
SP1(I1)=SFPI(I1)*1.E-6/G	00002390
SMO(I1)=SFMO(I1)*1.E-6	00002400

SM1(I1)=SFM1(I1)*1.E-6/G	00002410
650 CONTINUE	00002420
C	00002430
C	00002440
INITSW=1	00002450
TCAC=T+DT	00002460
RETURN	00002470
C	00002480
1000 FORMAT (I5,F20.10)	00002490
1010 FORMAT(36H ACC COMPENSATION INITIALIZATION ,	00002500
X/3X,8H DT ,3X,G16.8,3X,4H SEC,	00002510
X/3X,8H PRNTSW ,3X,G16.8,	00002520
X/3X,8H OUTSW ,3X,G16.8,	00002530
X/3X,8H OFILE ,3X,I15,	00002540
X/3X,8H SPARE ,3X,G16.8,	00002550
X/3X,8H MODPDT ,3X,G16.8,	00002560
X/3X,8H PRNTDT ,3X,G16.8,/,	00002570
X/3X,8H DELI ,3X,G16.8,9H GM CM**2,	00002580
X/18X,"X",18X,"Y",18X,"Z"	00002590
X/3X,8H BIAS ,3(3X,G16.8),8H MICRO G ,	00002600
X/3X,8H KII ,3(3X,G16.8), 13H MICRO G/G**2,	00002610
X/3X,8H MRC ,3X,G16.8, 6H GM CM,	00002620
X/3X,8H IXX ,3X,G16.8,9H GM CM**2,	00002630
X/3X,8H QABX ,3(3X,G16.8),	00002640
X/11X,3(3X,G16.8),/11X,3(3X,G16.8),	00002650
X/3X,8H QABY ,3(3X,G16.8),	00002660
X/11X,3(3X,G16.8),/11X,3(3X,G16.8),	00002670
X/3X,8H QABZ ,3(3X,G16.8),	00002680
X/11X,3(3X,G16.8),/11X,3(3X,G16.8))	00002690
1011 FORMAT (00002700
X/3X,8H QMIS ,3(3X,G16.8),	00002710
X/11X,3(3X,G16.8),/11X,3(3X,G16.8),	00002720
X/3X,8H SFM0 ,3(3X,G16.8), 4H PPM,	00002730
X/3X,8H SFM1 ,3(3X,G16.8), 7H PPM /G,	00002740
X/3X,8H SFP0 ,3(3X,G16.8), 4H PPM,	00002750
X/3X,8H SFP1 ,3(3X,G16.8), 7H PPM /G,	00002760
X/3X,8H RX ,3(3X,G16.8), 3H FT,	00002770
X/3X,8H RY ,3(3X,G16.8), 3H FT,	00002780
X/3X,8H RZ ,3(3X,G16.8), 3H FT,/,	00002790
1200 FORMAT(6X,25H ** CAC ** DV (FT/SEC) ,3G16.8,/,	00002800
END	00002810

C		00000010
C	03/10/78 DATE OF CURRENT MODULE	00000020
C		00000030
C	THE GYRO COMPENSATION MODULE COMPENSATES FOR ACCELEROMETER BIASES	00000040
C	AND ERRORS.	00000050
C		00000060
	SUBROUTINE GCOMP (T,IENDF,	00000070
X	DTHETO,DVA,DTHETZ)	00000080
C		00000090
	REAL BIAS(3)	00000100
	REAL BUF(17)	00000110
	REAL DATA(86)	00000120
	REAL DELI	00000130
	REAL DT	00000140
	REAL DTH(3)	00000150
	REAL DTHETG(3)	00000160
	REAL DTHETO(3)	00000170
	REAL DTHETZ(3)	00000180
	REAL DVA(3)	00000190
	REAL DVG(3)	00000200
	REAL DWO(3)	00000210
	REAL G	00000220
	REAL H	00000230
	REAL IXX	00000240
	REAL KI(3)	00000250
	REAL KO(3)	00000260
	REAL KS(3)	00000270
	REAL KII(3)	00000280
	REAL KIS(3)	00000290
	REAL KOS(3)	00000300
	REAL KIO(3)	00000310
	REAL KSS(3)	00000320
	REAL LAT	00000330
	REAL MODPOT	00000340
	REAL PBUF(16)	00000350
	REAL PDATA(20)	00000360
	REAL PRNTDT	00000370
	REAL QGBX(9)	00000380
	REAL QGBY(9)	00000390
	REAL QGBZ(9)	00000400
	REAL QHIS(9)	00000410
	REAL QOSG(9)	00000420
	REAL RE	00000430
	REAL SFMO(3)	00000440
	REAL SFM1(3)	00000450
	REAL SFP0(3)	00000460
	REAL SH0(3)	00000470
	REAL SH1(3)	00000480
	REAL SP0(3)	00000490
	REAL SP1(3)	00000500
	REAL SPARE1	00000510
	REAL SPARE2	00000520
	REAL SPP1(3)	00000530
	REAL T	00000540
	REAL T1	00000550
	REAL TFRMOO	00000560
	REAL TFRNT	00000570
	REAL WOOT(3)	00000580
	REAL WE	00000590
C		00000600

INTEGER IENDF	00000610
INTEGER IFILE	00000620
INTEGER INITSW	00000630
INTEGER OUTSW	00000640
INTEGER OFILE	00000650
INTEGER PFILE	00000660
C	00000670
EQUIVALENCE (DATA(1), DT)	00000680
EQUIVALENCE (DATA(2), PRNTSW)	00000690
EQUIVALENCE (DATA(3), OUTSW)	00000700
EQUIVALENCE (DATA(4), XFILE)	00000710
EQUIVALENCE (DATA(5), SPARE1)	00000720
EQUIVALENCE (DATA(6), SPARE2)	00000730
EQUIVALENCE (DATA(7), QMIS(1))	00000740
EQUIVALENCE (DATA(17), SFPD(1))	00000750
EQUIVALENCE (DATA(20), BIAS(1))	00000760
EQUIVALENCE (DATA(23), SFMD(1))	00000770
EQUIVALENCE (DATA(26), QGBX(1))	00000780
EQUIVALENCE (DATA(35), QGBY(1))	00000790
EQUIVALENCE (DATA(44), QGBZ(1))	00000800
EQUIVALENCE (DATA(53), KO(1))	00000810
EQUIVALENCE (DATA(56), KI(1))	00000820
EQUIVALENCE (DATA(59), KS(1))	00000830
EQUIVALENCE (DATA(62), KII(1))	00000840
EQUIVALENCE (DATA(65), H)	00000850
EQUIVALENCE (DATA(66), IXX)	00000860
EQUIVALENCE (DATA(67), KS3(1))	00000870
EQUIVALENCE (DATA(70), KIO(1))	00000880
EQUIVALENCE (DATA(73), KOS(1))	00000890
EQUIVALENCE (DATA(76), KIS(1))	00000900
EQUIVALENCE (DATA(79), DELI)	00000910
EQUIVALENCE (DATA(80), SFMI(1))	00000920
EQUIVALENCE (DATA(83), SFP1(1))	00000930
EQUIVALENCE (DATA(86), MODPOT)	00000940
C	00000950
EQUIVALENCE (PDATA(1), WE)	00000960
EQUIVALENCE (PDATA(2), RE)	00000970
EQUIVALENCE (PDATA(3), G)	00000980
EQUIVALENCE (PDATA(4), PRNTDT)	00000990
EQUIVALENCE (PDATA(5), PBUF(1))	00001000
C	00001010
DATA TPRMOD/0.0/	00001020
DATA TPRNT/0.0/	00001030
DATA IFILE/69/	00001040
DATA INITSW/0/	00001050
DATA K1 /0/	00001060
DATA PFILE /7/	00001070
C	00001080
IF (IENDF.EQ.1) RETURN	00001090
IF (INITSW.EQ.0) GO TO 500	00001100
IF (T.LT.TCGY-.0005) RETURN	00001110
C	00001120
C	00001130
DO 100 I=1,3	00001140
C	00001150
C COMPENSATE FOR SCALE FACTOR ERROR AND BIAS	00001160
C	00001170
IF (DTNETO(I).LT.0.) GOTO 90	00001180
C	00001190
C POSITIVE SCALE FACTOR ERROR	00001200

C	T1=1.+SP0(I)+SP1(I)*DTHETO(I)/DT	00001210
	GOTO 95	00001220
C		00001230
C	NEGATIVE SCALE FACTOR ERROR	00001240
C		00001250
90	T1=1.+SM0(I)+SM1(I)*DTHETO(I)/DT	00001260
95	DTHETO(I)=DTHETO(I)*T1+BIAS(I)*DT	00001270
C		00001280
100	CONTINUE	00001290
C		00001300
	DO 300 I1=1,3	00001310
C		00001320
C	TRANSFORM ACCELERATION FROM XYZ TO IOS	00001330
C		00001340
	IF (I1.NE.1) GOTO 210	00001350
C	X GYRO	00001360
C		00001370
	CALL MXV(QGBX,DVA,DVG)	00001380
	CALL MXV(QGBX,DTHETO,DTHETG)	00001390
	GOTO 230	00001400
C		00001410
210	IF (I1.NE.2) GOTO 220	00001420
C	Y GYRO	00001430
C		00001440
	CALL MXV(QGBY,DVA,DVG)	00001450
	CALL MXV(QGBY,DTHETO,DTHETG)	00001460
	GO TO 230	00001470
C	Z GYRO	00001480
C		00001490
220	CALL MXV(QGBZ,DVA,DVG)	00001500
	CALL MXV(QGBZ,DTHETO,DTHETG)	00001510
C		00001520
230	CONTINUE	00001530
C		00001540
C	COMPENSATE FOR ACCELERATION EFFECTS	00001550
C		00001560
	DTHETO(I1)=DTHETO(I1)+KI(I1)*DVG(1)+KO(I1)*DVG(2)+KS(I1)*	00001570
	X DVG(3)+(KOS(I1)*DVG(2)+DVG(3)+KSS(I1)*DVG(3)+DVG(3)+	00001580
	X KYI(I1)*DVG(1)+DVG(1)+KIS(I1)*DVG(1)+DVG(3)+KIO(I1)*DVG(1)+	00001590
	X DVG(2))/DT-DELI*DTHETO(1)+DTHETG(3)/H/DT	00001600
C		00001610
300	CONTINUE	00001620
	IF (K1.FQ.0) GOTO 410	00001630
	DO 350 I5=1,3	00001640
C		00001650
C	CALCULATE WDOT	00001660
C		00001670
	WDOT(I5)=DTHETO(I5)-DTH(I5)	00001680
350	CONTINUE	00001690
	CALL MXV(QO8G,WDOT,DWO)	00001700
C		00001710
	DO 400 I2=1,3	00001720
	DTH(I2)=DTHETO(I2)	00001730
	DTHETO(I2)=DTHETO(I2)+IXX*DWO(I2)/H/DT	00001740
400	CONTINUE	00001750
C		00001760
	GOTO 460	00001770
410	K1=1	00001780
	DO 450 I4=1,3	00001790
		00001800

DTH(I4)=DTHETO(I4)	00001810
450 CONTINUE	00001820
460 CONTINUE	00001830
C	00001840
C COMPENSATE FOR X,Y, AND Z MISSALIGNMENT	00001850
C	00001860
CALL MXV(QNIS,DTHETO,DHETZ)	00001870
C	00001880
C PRINT OUTPUT AND CONTROL	00001890
C	00001900
IF (PRNTDT.GT.0) GO TO 960	00001910
IF (MODPOT.EQ.0) GO TO 999	00001920
C	00001930
IF (T.LT.TPRMOD-.0005) GO TO 999	00001940
TPRMOD=TPRMOD+MODPOT	00001950
GO TO 970	00001960
C	00001970
960 CONTINUE	00001980
IF (T.LT.TPRNT-.0005) GO TO 999	00001990
TPRNT=TPRNT+PRNTUT	00002000
C	00002010
970 CONTINUE	00002020
IF (PRNTSM.LT.1.) GO TO 999	00002030
C	00002040
WRITE(OFIL,1200) DHETZ	00002050
C	00002060
C	00002070
999 CONTINUE	00002080
TGGY=T+DT	00002090
RETURN	00002100
C	00002110
C	00002120
C INITIALIZATION	00002130
C	00002140
500 CONTINUE	00002150
REWIND PFILE	00002160
REWIND IFILE	00002170
501 READ (IFILE,1000) IX,DATA(IX)	00002180
IF (EOF(IFILE)) 502,501	00002190
502 CONTINUE	00002200
503 READ (PFILE,1000) IX,PDATA(IX)	00002210
IF (EOF(PFILE)) 510,503	00002220
510 CONTINUE	00002230
C	00002240
REWIND PFILE	00002250
REWIND IFILE	00002260
OFIL=XFILE	00002270
C	00002280
WRITE (OFIL,1010) DT,PRNTSM,OUTSM,OFIL,MODPOT,PRNTDT,	00002290
X DIAS,SFPO,SFPO,GGON,GGDY,GGDZ,QNIS	00002300
WRITE (OFIL,1011) KI,KO,KS,KII,KIS,KOS,KSS,KIO,H,BELI,IXX	00002310
WRITE (OFIL,1012) SFH1,SFPI	00002320
C	00002330
DO 550 I1=1,3	00002340
SHI(I1)=SFH1(I1)*1.E-6	00002350
SPI(I1)=SFPI(I1)*1.E-6	00002360
SPDI(I1)=SFPO(I1)*1.E-6	00002370
SHDI(I1)=SFHO(I1)*1.E-6	00002380
550 CONTINUE	00002390
C	00002400

DO 529 I3=1,3	00002410
BIAS(I3)=BIAS(I3)*4.85E-6	00002420
KI(I3)=KI(I3)*4.85E-6/G	00002430
KO(I3)=KO(I3)*4.85E-6/G	00002440
KS(I3)=KS(I3)*4.85E-6/G	00002450
KII(I3)=KII(I3)*4.85E-6/G/G	00002460
KIS(I3)=KIS(I3)*4.85E-6/G/G	00002470
KOS(I3)=KOS(I3)*4.85E-6/G/G	00002480
KSS(I3)=KSS(I3)*4.85E-6/G/G	00002490
KIO(I3)=KIO(I3)*4.85E-6/G/G	00002500
529 CONTINUE	00002510
C	00002520
IXX=IXX*2.37E-6	00002530
DELI=DELI*2.37E-6	00002540
H=H*2.37E-6	00002550
QGBG(1)=QGBX(4)	00002560
QGBG(2)=QGBX(5)	00002570
QGBG(3)=QGBX(6)	00002580
QGBG(4)=QGBY(4)	00002590
QGBG(5)=QGBY(5)	00002600
QGBG(6)=QGBY(6)	00002610
QGBG(7)=QGBZ(4)	00002620
QGBG(8)=QGBZ(5)	00002630
QGBG(9)=QGBZ(6)	00002640
C	00002650
C	00002660
INITSW=1	00002670
TCGY=T+DT	00002680
RETURN	00002690
C	00002700
C	00002710
1000 FORMAT (I5,F20.10)	00002720
1010 FORMAT(36H GYR COMPENSATION INITIALIZATION	00002730
X/3X,0H DT ,3X,G16.8,3X,4H SEC,	00002740
X/3X,0H PRNTSW ,3X,G16.8,	00002750
X/3X,0H OUTSW ,3X,G16.8,	00002760
X/3X,0H OFILE ,3X,I15,	00002770
X/3X,0H MODPDT ,3X,G16.8,	00002780
X/3X,0H PRNTDT ,3X,G16.8,/,	00002790
X/10X,"X",10X,"Y",10X,"Z"	00002800
X/3X,0H BIAS ,3(3X,G16.8),7H DEG/HR,	00002810
X/3X,0H BFPD ,3(3X,G16.8), 4H PPM,	00002820
X/3X,0H BFM0 ,3(3X,G16.8), 4H PPM,/,	00002830
X/3X,0H QGBX ,3(3X,G16.8),	00002840
X/11X,3(3X,G16.8),/11X,3(3X,G16.8),/,	00002850
X/3X,0H QGBY ,3(3X,G16.8),	00002860
X/11X,3(3X,G16.8),/11X,3(3X,G16.8),/,	00002870
X/3X,0H QGBZ ,3(3X,G16.8),	00002880
X/11X,3(3X,G16.8),/11X,3(3X,G16.8),	00002890
X/3X,0H QMIS ,3(3X,G16.8),	00002900
X/11X,3(3X,G16.8),/11X,3(3X,G16.8),/)	00002910
1011 FORMAT(00002920
X/3X,0H KI ,3(3X,G16.8),9H DEG/HR/G,	00002930
X/3X,0H KO ,3(3X,G16.8),9H DEG/HR/G,	00002940
X/3X,0H KB ,3(3X,G16.8),9H DEG/HR/G,	00002950
X/3X,0H KII ,3(3X,G16.8),16H DEG/HR/G**2,	00002960
X/3X,0H KIS ,3(3X,G16.8),16H DEG/HR/G**2,	00002970
X/3X,0H KOS ,3(3X,G16.8),16H DEG/HR/G**2,	00002980
X/3X,0H KSS ,3(3X,G16.8),16H DEG/HR/G**2,	00002990
X/3X,0H KIO ,3(3X,G16.8),16H DEG/HR/G**2,	00003000

X/3X,8H H	,3X,G16.8,13H GM CM**2/SEC,	00003010
X/3X,8H DELT	,3X,G16.8,9H GM CM**2,	00003020
X/3X,8H IXX	,3X,G16.8,9H GM CM**2)	00003030
1012 FORMAT(00003040
X/3X,8H SFH1	,3(3X,G16.8),13H PPM /RAD/SEC,	00003050
X/3X,8H SFP1	,3(3X,G16.8),13H PPM /RAD/SEC,///)	00003060
1200 FORMAT(6X,27H ** CGY ** DTHETA (RAD) ,3G16.8,/)		00003070
END		00003080

```

C
C 3/15/78 DATE OF CURRENT MODULE
C
C 10/18/77 CHANGE FORMATS TO G16.8
C
C***** ***** RING LASER GYRO COMPENSATION MODULE ***** *****
C
C      SUBROUTINE GCAMP ( T,IENDF,DTHETO,DVA,
C                      DTHETZ )
C
C      .DTHETO - INCREMENTAL ROTATION GENERATED BY GYROS.
C      .DTHETZ - COMPENSATED INCREMENTAL ROTATION.
C      .DVA - NOT USED.
C
C      IMPLICIT REAL ( A-Z )
C      INTEGER I,IENDF,INITSW,IFILE,IX,J,K,OFIL,NP,PFILE
C      DIMENSION ABB(3),ANG(3),ANGRW(3),ANGWN(3),DATA(31),DTHETZ(3),
C      . DTHETO(3),DB(3),DTA(3),DTC(3),EXPD(3),EXPSF(3),KW(9),SF(3),
C      . SFTA(3),SFTC(3),VT1(3),DVA(3),PBUF(16),PDATA(20)
C
C      EQUIVALENCE (DATA(1), DT)
C      EQUIVALENCE (DATA(2), PRNTSW)
C      EQUIVALENCE (DATA(3), OUTSW)
C      EQUIVALENCE (DATA(4), XFILE)
C      EQUIVALENCE (DATA(5), SPARE1)
C      EQUIVALENCE (DATA(6), SPARE2)
C      EQUIVALENCE (DATA(7), MODPOT)
C      EQUIVALENCE (DATA(8), DB(1))
C      EQUIVALENCE (DATA(11), DTA(1))
C      EQUIVALENCE (DATA(14), DTC(1))
C      EQUIVALENCE (DATA(17), SFTA(1))
C      EQUIVALENCE (DATA(20), SFTC(1))
C      EQUIVALENCE (DATA(23), KW(1))
C
C      EQUIVALENCE (PDATA(1),WE)
C      EQUIVALENCE (PDATA(2),RE)
C      EQUIVALENCE (PDATA(3),G)
C      EQUIVALENCE (PDATA(4),PRNTDT)
C      EQUIVALENCE (PDATA(5),PBUF(1))
C
C      DATA DTR /.01745329251994330/
C      DATA INITSW /0/
C      DATA IFILE /69/
C      DATA OFIL /6/
C      DATA PFILE /7/
C      DATA TPRNT /0.0/
C      DATA TPRMOD /0.0/
C
C      IF (IENDF.EQ.1) RETURN
C      IF (INITSW.EQ.0) GO TO 500
C      IF (T.LT.TCGY-.0001) RETURN
C
C      -----BEGIN NORMAL COMPUTATIONS-----
C      CN*TRANSIENT DRIFT AND TRANSIENT SCALE FACTOR: DTA,SFTA.
C      100 DO 105 I = 1,3
C          IF ( ABS( DTA(I) ) .LT. 1.E-10 ) DTA(I) = 0.

```

```

      IF ( ABS( SFTA(I) ) .LT. 1.E-10 ) SFTA(I) = 0.
      DTA(I) = DTA(I)*EXPDI(I)
      SFTA(I) = SFTA(I)*EXPSFI(I)
      K=4*(I-1)+1
      KW(K) = SF(I) + SFTA(I)
105  CONTINUE
C**COMPUTE COMPENSATED INCREMENTAL ROTATION: DTHETZ.
      DO 110 I = 1,3
110  VT1(I) = DTHETO(I) - ( DB(I) + DTA(I) ) *DT
      CALL MXV ( KW,VT1,DTHETZ )
      DO 112 I = 1,3
112  DTHETZ(I) = VT1(I) - DTHETZ(I)
C-----END NORMAL COMPUTATIONS-----
C
C OUTPUT AND PRINT CONTROL
C
      IF ( PRNTDT.GT.0 ) GO TO 960
      IF ( MODPDT.EQ.0 ) GO TO 999
C
      IF ( T.LT.TPRMOD-.0005 ) GO TO 999
      TPRMOD=TPRMOD+MODPDT
      GO TO 970
C
960  CONTINUE
      IF ( T.LT.TPRNT-.0005 ) GO TO 999
      TPRNT=TPRNT+PRNTDT
C
970  CONTINUE
      IF ( PRNTSW.LT.1 ) GO TO 999
C
      WRITE(OFIL,1200) DTHETZ
C
999  CONTINUE
      TCGY=T+DT
      RETURN
C-----INITIALIZATION SECTION-----
C**READ THE INPUT DATA FILE: IFILE.
C      .DY      - SIMULATION TIME STEP          ( SEC )
C      .DB      - BIAS DRIFT                    ( DEG/HR )
C      .DTA     - DRIFT TRANSIENT AMPLITUDE      ( DEG/HR )
C      .DTC     - DRIFT TRANSIENT TIME CONSTANT  ( MIN )
C      .SFTA    - SCALE FACTOR TRANSIENT AMPLITUDE ( PPH )
C      .SFYC    - SCALE FACTOR TRANSIENT TIME CONSTANT ( MIN )
C      .KW      - SF AND GYRO ZA MISALIGNMENT MATRIX ( PPH )
C
500  CONTINUE
      REWIND PFILE
      REWIND IFILE
501  READ (IFILE,1000) IX,DATA(IX)
      IF ( EOF(IFILE) ) 503,501
502  CONTINUE
503  READ (PFILE,1000) IX,PDATA(IX)
      IF ( EOF(PFILE) ) 510,503
510  CONTINUE
      REWIND PFILE
      REWIND IFILE
      OFIL=XFILE

```


C	WRITE (OFILE,1010) DT,PRNTSW,OUTSW,OFILE,PRNTDT,MODPOT	00001210
	WRITE (OFILE,1011) DB,DTA,DTC,SFTA,SFTC,KW	00001220
C		00001230
C	C**SCALE THE INPUT DATA TO INTERNAL PROGRAM UNITS.	00001240
8	DO 10 I = 1,3	00001250
	DB(I) = DB(I)*DTR/3600.	00001260
	DTA(I) = DTA(I)*DTR/3600.	00001270
	DTC(I) = DTC(I)*60.	00001280
	SFTA(I) = SFTA(I)*1.E-06	00001290
	SFTC(I) = SFTC(I)*60.	00001300
	DTNETZ(I) = 0.	00001310
	DO 10 J = 1,3	00001320
	K=3*(I-1)+J	00001330
	KW(K) = KW(K)*1.E-06	00001340
10	CONTINUE	00001350
C		00001360
	DO 11 I = 1,3	00001370
	K=4*(I-1)+1	00001380
	SF(I) = KW(K)	00001390
	EXPDI(I) = 0.	00001400
	EXPSF(I) = 0.	00001410
	IF (DTC(I) .GT. 0.) EXPDI(I) = EXP(-DT/DTC(I))	00001420
	IF (SFTC(I) .GT. 0.) EXPSF(I) = EXP(-DT/SFTC(I))	00001430
11	CONTINUE	00001440
C		00001450
	INITSW=1	00001460
	TCGY=T*DT	00001470
	RETURN	00001480
	C**END INITIALIZATION.	00001490
C		00001500
C		00001510
	1000 FORMAT (I5,F20.10)	00001520
	1010 FORMAT(42H LASER GYRO COMPENSATION INITIALIZATION ,	00001530
	X/3X,6H DT ,3X,G16.8,3X,4H SEC,	00001540
	X/3X,6H PRNTC ,3X,G16.8,	00001550
	X/3X,6H OUTC ,3X,G16.8,	00001560
	X/3X,6H OFILE ,3X,I15,	00001570
	X/3X,6H PRNTDT ,3X,G16.8,3X,4H SEC,	00001580
	X/3X,6H MODPOT ,3X,G16.8,3X,4H SEC,/))	00001590
1011	FO IAT(00001600
	X/10X,"X",16X,"Y",16X,"Z",	00001610
	X/3X,6H DB ,3(3X,G16.8),7H DEG/HR,	00001620
	X/3X,6H DTA ,3(3X,G16.8),7H DEG/HR,	00001630
	Y/3X,6H DTC ,3(3X,G16.8),4H MIN,	00001640
	X/3X,6H SFTA ,3(3X,G16.8),4H PPH,	00001650
	Y/3X,6H SFTC ,3(3X,G16.8),4H MIN,	00001660
	X/3X,6H KW ,3(3X,G16.8),4H PPH,	00001670
	X/11X,3(3X,G16.8),/11X,3(3X,G16.8),/))	00001680
	1200 FORMAT(6X,27H ** CGY ** DYNETA (RAD) ,3(3X,G20.12))	00001690
	END	00001700
		00001710

C		00000010
C	05/30/78 DATE OF CURRENT MODULE	00000020
C		00000030
C	THE ATTITUDE AND ALGORITHM MODULE COMPUTES THE BODY TO INERTIAL	00000040
C	TRANSFORMATION, THE DIRECTION COSINE MATRIX, AND TRANSFORMS THE	00000050
C	INCREMENTAL VELOCITY COORDINATES.	00000060
C		00000070
	SUBROUTINE ALG (T,IENDF,DTHETI,DV,	00000080
	X DVN,DCM)	00000090
	REAL C1	00000100
	REAL C2	00000110
	REAL C3	00000120
	REAL COSLAT	00000130
	REAL COSLON	00000140
	REAL COSP	00000150
	REAL COSR	00000160
	REAL COSW	00000170
	REAL COSY	00000180
	REAL D	00000190
	REAL DATA(19)	00000200
	REAL DT	00000210
	REAL DTHETA(3)	00000220
	REAL DTHETI(3)	00000230
	REAL DTHP(3)	00000240
	REAL DTHPRE(3)	00000250
	REAL DTNRH	00000260
	REAL DV(3)	00000270
	REAL DVI(3)	00000280
	REAL DVN(3)	00000290
	REAL D1	00000300
	REAL G	00000310
	REAL ILAT	00000320
	REAL ILON	00000330
	REAL LATERR	00000340
	REAL LDELTA(4,4)	00000350
	REAL LOTH(3)	00000360
	REAL LONERR	00000370
	REAL HOOPDT	00000380
	REAL ORDER	00000390
	REAL QID(9)	00000400
	REAL QIP(9)	00000410
	REAL QPS(9)	00000420
	REAL QUAT(4)	00000430
	REAL QTEMP(4)	00000440
	REAL PDATA(20)	00000450
	REAL PITCH	00000460
	REAL PYERR	00000470
	REAL PRNTOT	00000480
	REAL RDTODG	00000490
	REAL RE	00000500
	REAL ROLL	00000510
	REAL ROLERR	00000520
	REAL SINLAT	00000530
	REAL SINLON	00000540
	REAL SINP	00000550
	REAL SINR	00000560
	REAL SINW	00000570
	REAL SINY	00000580
	REAL QORDCM	00000590
	REAL T	00000600

	REAL TALG	00000610
	REAL TPRMOD	00000620
	REAL TPRNT	00000630
	REAL UDELT(4,4)	00000640
	REAL UDT2	00000650
	REAL WE	00000660
	REAL WANDER	00000670
	REAL WANERR	00000680
	REAL YAW	00000690
	REAL YAWERR	00000700
C		00000710
	REAL DTH(9)	00000720
	REAL ATEH(9)	00000730
	REAL DCM(9)	00000740
	REAL Q1S,Q2S,Q3S	00000750
	REAL Q01,Q02,Q03	00000760
	REAL Q12,Q23,Q31	00000770
	REAL TFM(9)	00000780
	REAL DTEH(9)	00000790
	REAL SWET	00000800
	REAL CWET	00000810
C		00000820
	INTEGER NGDCM	00000830
	INTEGER IENDF	00000840
	INTEGER INITSW	00000850
	INTEGER IFILE	00000860
	INTEGER OFILE	00000870
	INTEGER OUTSW	00000880
	INTEGER PFILE	00000890
C		00000900
	EQUIVALENCE (DATA(1), DT)	00000910
	EQUIVALENCE (DATA(2), PRINTSW)	00000920
	EQUIVALENCE (DATA(3), CUTSW)	00000930
	EQUIVALENCE (DATA(4), XFILE)	00000940
	EQUIVALENCE (DATA(5), QORDCM)	00000950
	EQUIVALENCE (DATA(6), MODPOT)	00000960
	EQUIVALENCE (DATA(11), ORDER)	00000970
	EQUIVALENCE (DATA(12), DTHRM)	00000980
	EQUIVALENCE (DATA(13), DYSLOW)	00000990
	EQUIVALENCE (DATA(14), LATERR)	00001000
	EQUIVALENCE (DATA(15), LONERR)	00001010
	EQUIVALENCE (DATA(16), WANERR)	00001020
	EQUIVALENCE (DATA(17), PITERR)	00001030
	EQUIVALENCE (DATA(18), ROLERR)	00001040
	EQUIVALENCE (DATA(19), YAWERR)	00001050
C		00001060
	EQUIVALENCE (PDATA(1), WE)	00001070
	EQUIVALENCE (PDATA(2), HE)	00001080
	EQUIVALENCE (PDATA(3), G)	00001090
	EQUIVALENCE (PDATA(4), PRNTDT)	00001100
	EQUIVALENCE (PDATA(5), ILAT)	00001110
	EQUIVALENCE (PDATA(6), ILON)	00001120
	EQUIVALENCE (PDATA(7), WANDER)	00001130
	EQUIVALENCE (PDATA(8), ROLL)	00001140
	EQUIVALENCE (PDATA(10), PITCH)	00001150
	EQUIVALENCE (PDATA(11), YAW)	00001160
C		00001170
	DATA RDTODG /57.29577951/	00001180
	DATA YPRMOD/0.0/	00001190
	DATA YPRNT/0.0/	00001200

DATA INITSW/0/	00001210
DATA IFILE/70/	00001220
DATA K1/0/	00001230
DATA PFILE /7/	00001240
C	00001250
C	00001260
IF (IENDF.EQ.1) RETURN	00001270
IF (INITSW.EQ.0) GO TO 500	00001280
IF (T.LT.TALG-.001) RETURN	00001290
C	00001300
C STORE DCM AS DCMOLD FOR USE IN INTERPOLATED DCM, DCMID	00001310
C	00001320
DO 110 I2=1,9	00001330
DCMOLD(I2)=QIB(I2)	00001340
110 CONTINUE	00001350
C	00001360
DO 111 I3=1,3	00001370
DTHTA(I3)=DTHTI(I3)	00001380
DTHT(I3)=DTHTI(I3)	00001390
111 CONTINUE	00001400
C	00001410
CHOOSE QUATERNION OR DIRECTION COSINE MATRIX UPDATE CALCULATION	00001420
IF (NQDCM .EQ. 1) GO TO 600	00001430
C	00001440
BEGIN QUATERNION CALCULATION	00001450
C	00001460
UDEL(1,2)=-DTHTA(1)	00001470
UDEL(1,3)=-DTHTA(2)	00001480
UDEL(1,4)=-DTHTA(3)	00001490
UDEL(2,1)=DTHTA(1)	00001500
UDEL(2,3)=DTHTA(3)	00001510
UDEL(2,4)=-DTHTA(2)	00001520
UDEL(3,1)=DTHTA(2)	00001530
UDEL(3,2)=-DTHTA(3)	00001540
UDEL(3,4)=DTHTA(1)	00001550
UDEL(4,1)=DTHTA(3)	00001560
UDEL(4,2)=DTHTA(2)	00001570
UDEL(4,3)=-DTHTA(1)	00001580
IF (NORDER.NE.1) GOTO 140	00001590
C	00001600
DO 120 I2=1,4	00001610
C FIRST ORDER QUATERNION	00001620
QTEMP(I2)=QUAT(I2)*2.	00001630
C	00001640
DO 120 J2=1,4	00001650
IF (I2.EQ.J2) GOTO 120	00001660
QTEMP(I2)=UDEL(I2,J2)*QUAT(J2)+QTEMP(I2)	00001670
120 CONTINUE	00001680
C	00001690
DO 130 I3=1,4	00001700
QUAT(I3)=QTEMP(I3)/2.	00001710
130 CONTINUE	00001720
C	00001730
GOTO 200	00001740
C	00001750
140 CONTINUE	00001760
UDTH=-10DTHTA(1)+DTHTA(1)+DTHTA(2)+DTHTA(2)+DTHTA(3)+	00001770
X DTHTA(3))	00001780
C	00001790
IF (NORDER.NE.2) GOTO 170	00001800

145	K1=1	00001810
C	SECOND ORDER QUATERNION	00001820
	C1=2.*UDTH2/4.	00001830
C		00001840
	DO 150 I4=1,4	00001850
	QTEMP(I4)=QUAT(I4)*C1	00001860
C		00001870
	DO 150 J4=1,4	00001880
	IF (J4.EQ.I4) GOTO 150	00001890
	QTEMP(I4)=UDELTA(I4,J4)*QUAT(J4)+QTEMP(I4)	00001900
150	CONTINUE	00001910
C		00001920
	DO 160 I5=1,4	00001930
	QUAT(I5)=QTEMP(I5)/2.	00001940
160	CONTINUE	00001950
C		00001960
	GOTO 200	00001970
170	IF(NORDER.NE.3) GOTO 300	00001980
C		00001990
C	THIRD ORDER QUATERNION	00002000
	IF (K1.EQ.0) GOTO 145	00002010
	LDTH(1)=DTHPRE(2)*DTHETA(3)-DTHPRE(3)*DTHETA(2)	00002020
	LDTH(2)=DTHPRE(3)*DTHETA(1)-DTHPRE(1)*DTHETA(3)	00002030
	LDTH(3)=DTHPRE(1)*DTHETA(2)-DTHPRE(2)*DTHETA(1)	00002040
	LDELTA(1,2)=-LDTH(1)	00002050
	LDELTA(1,3)=-LDTH(2)	00002060
	LDELTA(1,4)=-LDTH(3)	00002070
	LDELTA(2,1)=LDTH(1)	00002080
	LDELTA(2,3)=LDTH(3)	00002090
	LDELTA(2,4)=-LDTH(2)	00002100
	LDELTA(3,1)=LDTH(2)	00002110
	LDELTA(3,2)=-LDTH(3)	00002120
	LDELTA(3,4)=LDTH(1)	00002130
	LDELTA(4,1)=LDTH(3)	00002140
	LDELTA(4,2)=LDTH(2)	00002150
	LDELTA(4,3)=-LDTH(1)	00002160
	C2=1.+1./8.*UDTH2	00002170
	C3=1./2.+1./48.*UDTH2	00002180
:		00002190
	DO 180 I6=1,4	00002200
	QTEMP(I6)=C2*QUAT(I6)	00002210
:		00002220
	DO 180 J6=1,4	00002230
	IF (J6.EQ.I6) GOTO 180	00002240
	QTEMP(I6)=(C3*UDELTA(I6,J6)+LDELTA(I6,J6)/24.)*QUAT(J6)+QTEMP(I6)	00002250
180	CONTINUE	00002260
C		00002270
	DO 190 I7=1,4	00002280
	QUAT(I7)=QTEMP(I7)	00002290
190	CONTINUE	00002300
C		00002310
C	END QUATERNION CALCULATION	00002320
C		00002330
200	CONTINUE	00002340
C	NORMALIZE QUATERNION	00002350
	IF (1.TLT.TNORM-.001)GOTO 900	00002360
	Q=QUAT(1)*QUAT(1)+QUAT(2)*QUAT(2)+QUAT(3)*QUAT(3)	00002370
	+QUAT(4)*QUAT(4)	00002380
	IF (D.EQ.1.) GOTO 900	00002390
	D1=SQRT(10)	00002400

C		00002410
	DO 350 J9=1,4	00002420
850	QUAT(J9)=QUAT(J9)/D1	00002430
C		00002440
	TNORM=T+DTNRM	00002450
900	CONTINUE	00002460
C		00002470
C	CONVERSION FROM QUATERNION TO DIRECTION COSINE MATRIX :	00002480
C		00002490
	Q1S=QUAT(2)*QUAT(2)	00002500
	Q2S=QUAT(3)*QUAT(3)	00002510
	Q3S=QUAT(4)*QUAT(4)	00002520
	Q01=QUAT(1)*QUAT(2)	00002530
	Q02=QUAT(1)*QUAT(3)	00002540
	Q03=QUAT(1)*QUAT(4)	00002550
	Q12=QUAT(2)*QUAT(3)	00002560
	Q23=QUAT(3)*QUAT(4)	00002570
	Q31=QUAT(4)*QUAT(2)	00002580
		00002590
	DCM(1)=1.0-2.0*(Q2S+Q3S)	00002600
	DCM(2)=2.0*(Q12-Q03)	00002610
	DCM(3)=2.0*(Q31+Q02)	00002620
	DCM(4)=2.0*(Q12+Q03)	00002630
	DCM(5)=1.0-2.0*(Q3S+Q1S)	00002640
	DCM(6)=2.0*(Q23-Q01)	00002650
	DCM(7)=2.0*(Q31-Q02)	00002660
	DCM(8)=2.0*(Q23+Q01)	00002670
	DCM(9)=1.0-2.0*(Q1S+Q2S)	00002680
	GO TO 700	00002690
C		00002700
	600 CONTINUE	00002710
C		00002720
C	DIRECTION COSINE MATRIX UPDATE CALCULATION	00002730
C		00002740
	IF (NORDER.LT. 1) GO TO 300	00002750
C		00002760
C	FIRST ORDER UPDATE	00002770
C		00002780
	DTM(1)= 1.0	00002790
	DTM(5)= 1.0	00002800
	DTM(9)= 1.0	00002810
	DTM(2)=-DTHEA(3)	00002820
	DTM(3)= DTHEA(2)	00002830
	DTM(4)= DTHEA(3)	00002840
	DTM(6)=-DTHEA(1)	00002850
	DTM(7)=DTHEA(2)	00002860
	DTM(8)= DTHEA(1)	00002870
C		00002880
C	SECOND ORDER UPDATE	00002890
C		00002900
	IF (NORDER .LT. 2) GO TO 620	00002910
	LOTH(1)=DTHEA(1)**2	00002920
	LOTH(2)=DTHEA(2)**2	00002930
	LOTH(3)=DTHEA(3)**2	00002940
	DTM(1)+DTM(1)-0.5*(LOTH(2)+LOTH(3))	00002950
	DTM(5)+DTM(5)-0.5*(LOTH(3)+LOTH(1))	00002960
	DTM(9)+DTM(9)-0.5*(LOTH(1)+LOTH(2))	00002970
	DTM(2)+DTM(2)+0.5*DTHEA(1)*DTHEA(2)	00002980
	DTM(3)+DTM(3)+0.5*DTHEA(3)*DTHEA(1)	00002990
	DTM(6)+DTM(6)+0.5*DTHEA(2)*DTHEA(3)	00003000

DTM(4)=DTM(4)+0.5*DTHEA(1)*DTHEA(2)	00003010
DTM(7)=DTM(7)+0.5*DTHEA(3)*DTHEA(1)	00003020
DTM(8)=DTM(8)+0.5*DTHEA(2)*DTHEA(3)	00003030
620 CONTINUE	00003040
C	00003050
C THIRD ORDER UPDATE	00003060
C	00003070
IF (NORDER .LT. 3) GO TO 630	00003080
D=0.1666666666666666*(LDTH(1)+LDTH(2)+LDTH(3))	00003090
C THIRD ORDER TERMS SET TO ZERO DUE TO PROFGEN INFINITE ANGULAR ACCEL	00003100
C1=0.0	00003110
C2=0.0	00003120
C3=0.0	00003130
C EXPRESSIONS FOR THIRD ORDER TERMS	00003140
C C1=(DTHEA(3)*DTHPRE(2)-DTHEA(2)*DTHPRE(3))*0.6333333333333333	00003150
C C2=(-DTHEA(3)*DTHPRE(1)+DTHEA(1)*DTHPRE(3))*0.6333333333333333	00003160
C C3=(DTHEA(2)*DTHPRE(1)-DTHEA(1)*DTHPRE(2))*0.6333333333333333	00003170
C	00003180
DTM(2)=DTM(2)+D*DTHEA(3)-C3	00003190
DTM(4)=DTM(4)-D*DTHEA(3)+C3	00003200
DTM(3)=DTM(3)-D*DTHEA(2)+C2	00003210
DTM(7)=DTM(7)+D*DTHEA(2)-C2	00003220
DTM(6)=DTM(6)+D*DTHEA(1)-C1	00003230
DTM(8)=DTM(8)-D*DTHEA(1)+C1	00003240
630 CONTINUE	00003250
IF (NORDER .GT. 3) GO TO 300	00003260
992 FORMAT(10X,6H TEST ,3G16.8,/24X,3G16.8,/24X,3G16.8,/))	00003270
C WRITE (OFIL,992)DTM	00003280
C WRITE (OFIL,992)QIB	00003290
C CALL HXN(QIB,DTM,DCH)	00003300
C WRITE (OFIL,992)DCH	00003310
C	00003320
C NORMALIZE DIRECTION COSINE MATRIX	00003330
C	00003340
IF (Y .LT. YNORM-.001) GO TO 700	00003350
DO 710 I1=1,9	00003360
DTM(I1)=DCH(I1)	00003370
710 CONTINUE	00003380
CALL HXN(DTM,DCH,ATEM)	00003390
ATEM(1)=ATEM(1)-1.0	00003400
ATEM(5)=ATEM(5)-1.0	00003410
ATEM(9)=ATEM(9)-1.0	00003420
CALL HXN(DCH,ATEM,DTM)	00003430
DO 720 I1=1,9	00003440
DCH(I1)=DCH(I1)-0.5*DTM(I1)	00003450
720 CONTINUE	00003460
YNORM=Y+DTNRM	00003470
700 CONTINUE	00003480
C	00003490
DO 730 I1=1,9	00003500
QIB(I1)=DCH(I1)	00003510
730 CONTINUE	00003520
C	00003530
C END OF DCH UPDATE	00003540
C	00003550
800 CONTINUE	00003560
C WRITE (OFIL,992)DCH	00003570
C	00003580
C END OF QUATERNION OR DCH UPDATE	00003590
C	00003600

C	FORM APPROXIMATE MIDPOINT DCM, DCMID	00003610
C		00003620
	DO 192 K1=1.9	00003630
	DCMID(K1)=0.5*(DCM(K1)+DCMOLD(K1))	00003640
192	CONTINUE	00003650
C		00003660
C	CALCULATE ROTATED DV	00003670
C		00003680
	CALL MXV(DCMID,DV,DVI)	00003690
C		00003700
C	END DV CALCULATION	00003710
C		00003720
	DVN(1)=DVN(1)+DVI(3)	00003730
	DVN(2)=DVN(2)+DVI(1)	00003740
	DVN(3)=DVN(3)+DVI(2)	00003750
C		00003760
	SWET=SIN(WE* π)	00003770
	CWET=COS(WE* π)	00003780
		00003790
	DO 191 K9=1.9	00003800
	DTM(K9)=DCM(K9)	00003810
191	CONTINUE	00003820
	TFM(1)=SWET	00003830
	TFM(2)=0.0	00003840
	TFM(3)=CWET	00003850
	TFM(4)=CWET	00003860
	TFM(5)=0.0	00003870
	TFM(6)=-SWET	00003880
	TFM(7)=0.0	00003890
	TFM(8)=1.0	00003900
	TFM(9)=0.0	00003910
C		00003920
	CALL MXN(TFM,DTM,DCM)	00003930
C	WRITE (OFILE,992)DCM	00003940
C		00003950
C		00003960
C	DCM IS BODY TO EARTH FIXED TRANSFORMATION	00003970
C		00003980
C		00003990
	IF(INORDER.NE.3) GO TO 950	00004000
	DYHPRE(1)=DYHETA(1)	00004010
	DYHPRE(2)=DYHETA(2)	00004020
	DYHPRE(3)=DYHETA(3)	00004030
950	CONTINUE	00004040
C		00004050
C	OUTPUT AND PRINT CONTROL	00004060
C		00004070
	IF (IPRINTD.GT.0) GO TO 960	00004080
	IF (IHOPOT.EQ.0) GO TO 999	00004090
C		00004100
	IF (T.LT.TPRNCD-.0005) GO TO 999	00004110
	TPRNCDD=TPRNCDD+IHOPOT	00004120
	GO TO 970	00004130
C		00004140
960	CONTINUE	00004150
	IF (T.LT.TPRNT-.0005) GO TO 999	00004160
	TPRNT=TPRNT+IPRINTD	00004170
C		00004180
970	CONTINUE	00004190
	IF (IPRNTSW.LT.1.) GO TO 999	00004200

WRITE (OFILE,1300) DVN	00004210
IF (NGUCH.EQ. 0) WRITE (OFILE,1500) QUAT	00004220
WRITE (OFILE,1400) DCH	00004230
C	00004240
C	00004250
999 CONTINUE	00004260
TALG=T + DT	00004270
RETURN	00004280
C	00004290
C	00004300
ERROR RETURN FROM ALO MODULE	00004310
C	00004320
300 PRINT 320	00004330
320 FORMAT (3X,"ORDER NOT PROPERLY SPECIFIED")	00004340
IENDF=1	00004350
RETURN	00004360
C	00004370
C	00004380
ALGORITHM INITIALIZATION	00004390
500 CONTINUE	00004400
REWIND IFILE	00004410
501 READ (IFILE,1000) IX,DATA(IX)	00004420
IF (EOF(IFILE)) 502,501	00004430
502 CONTINUE	00004440
503 READ(PFILE,1000) IX,PDATA(IX)	00004450
IF (EOF(PFILE)) 504,503	00004460
504 CONTINUE	00004470
C	00004480
REWIND PFILE	00004490
REWIND IFILE	00004500
OFILE=XFILE	00004510
C	00004520
DO 501 I=1,3	00004530
DVN(I)=0	00004540
DTHPR(I)=0.0	00004550
521 CONTINUE	00004560
C	00004570
TNORM=T+DTHPR	00004580
NDCH=GDCH	00004590
NORDER=ORDER	00004600
COSP=COS(PITCH+PITERR/ROTODG)	00004610
COSY=COS(YAW+YAWERR/ROTODG)	00004620
COSR=COS(ROLL+ROLLERR/ROTODG)	00004630
COSLAT=COS(ILAT+LATERR/ROTODG)	00004640
COSLON=COS(ILON+LONERR/ROTODG)	00004650
COSW=COS(WANDER+WANDERR/ROTODG)	00004660
SINP=SIN(PITCH+PITERR/ROTODG)	00004670
SINY=SIN(YAW+YAWERR/ROTODG)	00004680
SINR=SIN(ROLL+ROLLERR/ROTODG)	00004690
SINLAT=SIN(ILAT+LATERR/ROTODG)	00004700
SINLON=SIN(ILON+LONERR/ROTODG)	00004710
SINW=SIN(WANDER+WANDERR/ROTODG)	00004720
QPC11=COSP*SINY	00004730
QPC12=SINP*SINR*SINY-COSY-COSR	00004740
QPC13=COSR*SINP*SINY+COSY*SINR	00004750
QPC14=COSP*COSY	00004760
QPC15=SINR*SINP*COSY+COSR*SINY	00004770
QPC16=-COSR*SINP*COSY-SINR*SINY	00004780
QPC17=SINP	00004790
QPC18=COSP*SINR	00004800
QPC19=COSP*COSR	

	QIP(1)=COSLON*COSW-SINLON*SINW*SINLAT	00004810
	QIP(2)=-SINW*COSLON-COSW*SINLON*SINLAT	00004820
	QIP(3)=COSLAT*SINLON	00004830
	QIP(4)=SINW*COSLAT	00004840
	QIP(5)=COSW*COSLAT	00004850
	QIP(6)=SINLAT	00004860
	QIP(7)=-COSW*SINLON-SINW*SINLAT*COSLON	00004870
	QIP(8)=SINW*SINLON-SINLAT*COSLON*COSW	00004880
	QIP(9)=COSLAT*COSLON	00004890
	CALL HXN(QIP,QPB,QIB)	00004900
	WRITE (OFIL,993)QIB	00004910
993	FORMAT(" QIB=",4X,3F15.8,2(/,8X,3F15.8))	00004920
	QUAT(1)=SGRT((1.+QIB(1)+QIB(5)+QIB(9))/4.)	00004930
	QUAT(2)=(QIB(8)-QIB(6))/(4.*QUAT(1))	00004940
	QUAT(3)=(QIB(3)-QIB(7))/(4.*QUAT(1))	00004950
	QUAT(4)=(QIB(4)-QIB(2))/(4.*QUAT(1))	00004960
C		00004970
	IF (NQDCH.EQ. 0) GO TO 995	00004980
	DD 994 I1=1,4	00004990
	QUAT(I1)=0.0	00005000
994	CONTINUE	00005010
995	CONTINUE	00005020
C		00005030
	WRITE (OFIL,1010) DT,PRYSW,OUTSW,OFIL,QORDCH,MODPOT,PRNTDT,	00005040
	X QUAT,NORERR,DTRRH,LATERR,LONERR,MANERR,PITERR,ROLERR,YAWERR	00005050
C		00005060
	DD 531 I3=1,3	00005070
	DTRH(I3)=0.	00005080
531	CONTINUE	00005090
C		00005100
	INITSW=1	00005110
	TALG=Y+DT	00005120
	RETURN	00005130
C		00005140
C		00005150
	1000 FORMAT (15,F20.10)	00005160
	1010 FORMAT(30H ALGORITHM INITIALIZATION	00005170
	X/3X,0H DT ,3X,G16.8,3X,0H SEC,	00005180
	X/3X,0H PRYSW ,3X,G16.8,	00005190
	X/3X,0H OUTSW ,3X,G16.8,	00005200
	X/3X,0H OFIL ,3X,I15,	00005210
	X/3X,0H QORDCH ,3X,G16.8,	00005220
	X/3X,0H MODPOT ,3X,G16.8,	00005230
	X/3X,0H PRNTDT ,3X,G16.8,	00005240
	X/3X,0H QUAT ,4(1X,G16.8),	00005250
	X/3X,0H ORDR ,3X,I5,	00005260
	X/3X,0H DTRRH ,3X,G16.8,	00005270
	X/3X,0H LATERR ,3X,G16.8,4H DEG,	00005280
	X/3X,0H LONERR ,3X,G16.8,4H DEG,	00005290
	X/3X,0H MANERR ,3X,G16.8,4H DEG,	00005300
	X/3X,0H PITERR ,3X,G16.8,4H DEG,	00005310
	X/3X,0H ROLERR ,3X,G16.8,4H DEG,	00005320
	X/3X,0H YAWERR ,3X,G16.8,4H DEG,/))	00005330
C		00005340
	1200 FORMAT(6X,25H ** ALG ** DVI(FT/SEC) ,3G16.8,/))	00005350
	1300 FORMAT(6X,25H ** ALG ** DVH(FT/SEC) ,3G16.8,/))	00005360
	1400 FORMAT(10X,6H DCH ,3G16.8,/,24X,3G16.8,/,24X,3G16.8,/))	00005370
	1500 FORMAT(18X,6H QUAT ,4G16.8,/))	00005380
	END	00005390

C		00000010
C	05/10/78 DATE OF CURRENT MODULE	00000020
C		00000030
C	THE LOCAL LEVEL NAVIGATION MODULE TRANSFORMS THE INCREMENTAL	00000040
C	VELOCITIES TO LOCAL LEVEL HANDED AZIMUTH COORDINATES AND COMPUTES	00000050
C	THE POSITION, VELOCITY, AND ATTITUDE IN THIS FRAME USING ALGORITHMS	00000060
C	SIMILAR TO THE UPGRADED NUMSIM ALGORITHMS (R-977,VOL II,SEC 2.4.4)	00000070
C		00000080
	SUBROUTINE LLN (T,IENDF,DVI,ALTO,DCM,	00000090
	X NAVLAT,NAVLO,NAVV,NAVH,NAVP,NAVR,NAVHD)	00000100
C		00000110
	REAL A(3,3)	00000120
	REAL ALF	00000130
	REAL ALTERR	00000140
	REAL ALTO	00000150
	REAL ATA(3,3)	00000160
	REAL ATEMP(3,3)	00000170
	REAL CALF	00000180
	REAL CGDL	00000190
	REAL CCSLON	00000200
	REAL COSLT	00000210
	REAL CWET	00000220
	REAL DATA(15)	00000230
	REAL DROLL	00000240
	REAL DPITCH	00000250
	REAL DYAW	00000260
	REAL DVC(3)	00000270
	REAL DVE(3)	00000280
	REAL DVI(3)	00000290
	REAL DT	00000300
	REAL G	00000310
	REAL GR(3)	00000320
	REAL IAX	00000330
	REAL IAY	00000340
	REAL IAZ	00000350
	REAL ILAT	00000360
	REAL ILON	00000370
	REAL IH	00000380
	REAL INAVLA	00000390
	REAL INAVLO	00000400
	REAL ITEMP1	00000410
	REAL ITEMP2	00000420
	REAL ITEMP3	00000430
	REAL ITEMP4	00000440
	REAL ITEMP5	00000450
	REAL IV(3)	00000460
	REAL LATERR	00000470
	REAL LONERR	00000480
	REAL MODFDT	00000490
	REAL NAVH	00000500
	REAL NAVLAT	00000510
	REAL NAVLO	00000520
	REAL NAVV(3)	00000530
	REAL PDATA(20)	00000540
	REAL PITCH	00000550
	REAL PRNTDT	00000560
	REAL RE	00000570
	REAL RDTODG	00000580
	REAL RHO(3)	00000590
	REAL ROLL	00000600

REAL SALF
 REAL SPARE1
 REAL SPARE2
 REAL SWET
 REAL SINLON
 REAL SINLT
 REAL THG(3)
 REAL TPRMOD
 REAL TPRNT
 REAL V(3)
 REAL VERR(3)
 REAL WE
 REAL WET
 REAL WONDER
 REAL WXV(3)
 REAL YAW

C

REAL NAVP
 REAL NAVR
 REAL NAVHD
 REAL DEN
 REAL DCM(9)
 REAL AV(9)
 REAL OTEM(9)
 REAL OV(3)
 REAL HV(3)
 REAL XV(3)
 REAL DELH
 REAL CVD1
 REAL CVD2
 REAL CVD3
 REAL VCMF
 REAL SSLAT
 REAL XSSLAT
 REAL XCALF
 REAL XSALF
 REAL THY
 REAL THZ
 REAL M2DT(3,3)
 REAL M2DT02(3,3)
 REAL XA(3,3)
 REAL H
 REAL HB
 REAL OHB
 REAL XHB
 REAL OALF
 REAL DALF

C

INTEGER NORTH
 INTEGER NUP
 INTEGER PFILE
 INTEGER OFILE

C

EQUIVALENCE (DATA(1), DT)
 EQUIVALENCE (DATA(2), PRNTSW)
 EQUIVALENCE (DATA(3), OUTSW)
 EQUIVALENCE (DATA(4), XFILE)
 EQUIVALENCE (DATA(5), SPARE1)
 EQUIVALENCE (DATA(6), MODPDT)
 EQUIVALENCE (DATA(7), ALTERR)

00000610
 00000620
 00000630
 00000640
 00000650
 00000660
 00000670
 00000680
 00000690
 00000700
 00000710
 00000720
 00000730
 00000740
 00000750
 00000760
 00000770
 00000780
 00000790
 00000800
 00000810
 00000820
 00000830
 00000840
 00000850
 00000860
 00000870
 00000880
 00000890
 00000900
 00000910
 00000920
 00000930
 00000940
 00000950
 00000960
 00000970
 00000980
 00000990
 00001000
 00001010
 00001020
 00001030
 00001040
 00001050
 00001060
 00001070
 00001080
 00001090
 00001100
 00001110
 00001120
 00001130
 00001140
 00001150
 00001160
 00001170
 00001180
 00001190
 00001200

	EQUIVALENCE (DATA(8),VERR(1))	00001210
	EQUIVALENCE (DATA(11),LATERR)	00001220
	EQUIVALENCE (DATA(12),LONERR)	00001230
	EQUIVALENCE (DATA(13),CVD1)	00001240
	EQUIVALENCE (DATA(14),CVD2)	00001250
	EQUIVALENCE (DATA(15),CVD3)	00001260
C		00001270
	EQUIVALENCE (PDATA(1), WE)	00001280
	EQUIVALENCE (PDATA(2), RE)	00001290
	EQUIVALENCE (PDATA(3), G)	00001300
	EQUIVALENCE (PDATA(4), PRNTDT)	00001310
	EQUIVALENCE (PDATA(5), ILAT)	00001320
	EQUIVALENCE (PDATA(6), ZLON)	00001330
	EQUIVALENCE (PDATA(7), WONDER)	00001340
	EQUIVALENCE (PDATA(8), IH)	00001350
	EQUIVALENCE (PDATA(9), ROLL)	00001360
	EQUIVALENCE (PDATA(10), PITCH)	00001370
	EQUIVALENCE (PDATA(11), YAW)	00001380
	EQUIVALENCE (PDATA(12), DROLL)	00001390
	EQUIVALENCE (PDATA(13), DPITCH)	00001400
	EQUIVALENCE (PDATA(14), DYAW)	00001410
	EQUIVALENCE (PDATA(15), IV(1))	00001420
	EQUIVALENCE (PDATA(16), IV(2))	00001430
	EQUIVALENCE (PDATA(17), IV(3))	00001440
	EQUIVALENCE (PDATA(18), IAX)	00001450
	EQUIVALENCE (PDATA(19), IAY)	00001460
	EQUIVALENCE (PDATA(20), IAZ)	00001470
C		00001480
	DATA RDTODG /57.29577951/	00001490
	DATA TPRMOD/0.0/	00001500
	DATA TPRNT/0.0/	00001510
	DATA IFILE/80/	00001520
	DATA INITSW/0/	00001530
	DATA PFILE /7/	00001540
C		00001550
C		00001560
	IF (IENDF.EQ.1) RETURN	00001570
	IF (INITSW.EQ.0) GO TO 500	00001580
	IF (T.LT.TLLN-.0005) RETURN	00001590
C		00001600
C	TRANSFORM FROM INERTIAL TO EARTHFIXED (AT MID COMP CYCLE)	00001610
C		00001620
	WET=WE*(T-0.5*DT)	00001630
	SWET=SIN(WET)	00001640
	CWET=COS(WET)	00001650
C		00001660
	DVE(1)=CWET*DVI(1)+SWET*DVI(2)	00001670
	DVE(2)=-SWET*DVI(1)+CWET*DVI(2)	00001680
	DVE(3)=DVI(3)	00001690
C		00001700
C	TRANSFORM FROM EARTHFIXED TO COMPUTATIONAL (MID-COMP CYCLE)	00001710
C		00001720
	DO 10 I1=1,3	00001730
	DVC(I1)= XA(1,I1) * DVE(1)+ XA(2,I1) * DVE(2)+ XA(3,I1) * DVE(3)	00001740
10	CONTINUE	00001750
C		00001760
C	UPDATE VELOCITY (UEN FOR ALF=0)	00001770
C		00001780
	DO 20 I2=1,3	00001790
	OV(I2)=V(I2)	00001800

V(I2)=V(I2)+DVC(I2)+MXV(I2)+GR(I2)*DT	90001810
20 CONTINUE	00001820
C	00001830
C ADD VERTICAL DAMPING TERM	00001840
C	00001850
V(1)=V(1)+(VDMP+CVD2*DELH)*DT	00001860
C	00001870
C AVERAGE OLD & NEW V(I) TO OBTAIN INTERPOLATED VALUE, HV(I)	00001880
C AND EXTRAPOLATE TO OBTAIN XV(I)	00001890
C	00001900
DO 25 I=1,3	00001910
XV(I) = 1.5*V(I) - 0.5*OV(I)	00001920
HV(I) = 0.5*(V(I) + OV(I))	00001930
25 CONTINUE	00001940
C	00001950
C COMPUTE ANGULAR VELOCITY (RHO)	00001960
C	00001970
CALL ANGVEL(HV,XCALF,XSALF,XSSLAT,XH,RHO)	00001980
C WRITE (OFIL,551) RHO	00001990
C	00002000
C UPDATE TRANSFORMATION FROM COMPUTATIONAL TO EARTH FIXED	00002010
C	00002020
THY = RHO(2)*DT	00002030
THZ = RHO(3)*DT	00002040
CALL AUP(THY,THZ,M2DT)	00002050
C WRITE (OFIL,551) M2DT	00002060
CALL MM(A,M2DT,ATEMP)	00002070
C	00002080
DO 40 M3=1,3	00002090
DO 40 M4=1,3	00002100
40 A(M3,M4)=ATEMP(M3,M4)	00002110
C WRITE (OFIL,551) A	00002120
NUP = NUP + 1	00002130
C	00002140
C IS IT TIME TO ORTHONORMALIZE A?	00002150
C	00002160
IF (NUP .LT. NORTH) GO TO 80	00002170
NUP=0	00002180
C	00002190
C ORTHONORMALIZE TRANSFORMATION	00002200
C	00002210
DO 50 I4=1,3	00002220
DO 50 I5=I4,3	00002230
ATA(I4,I5)=A(1,I4)*A(1,I5)+A(2,I4)*A(2,I5)+A(3,I4)*A(3,I5)	00002240
ATA(I5,I4)=ATA(I4,I5)	00002250
50 CONTINUE	00002260
C	00002270
C WRITE (OFIL,551) ATA	00002280
ATA(1,1)=ATA(1,1)-1.	00002290
ATA(2,2)=ATA(2,2)-1.	00002300
ATA(3,3)=ATA(3,3)-1.	00002310
C	00002320
DO 70 I6=1,3	00002330
DO 70 I7=1,3	00002340
ATEMP(I6,I7)=A(I6,I7)-.5*(A(I6,1)*ATA(1,I7)	00002350
X +A(I6,2)*ATA(2,I7)+A(I6,3)*ATA(3,I7))	00002360
70 CONTINUE	00002370
80 CONTINUE	00002380
C	00002390
C RESTORE MATRIX A ROW WISE AS VECTOR AV	00002400

C	II=0	00002410
	DO 90 I8=1,3	00002420
	DO 90 I9=1,3	00002430
	II=II+1	00002440
	A(I8,I9)=ATEMP(I8,I9)	00002450
	AV(II)=A(I8,I9)	00002460
90	CONTINUE	00002470
C		00002480
C	WRITE (OFIL,551) A	00002490
C		00002500
C	EXTRAPOLATE A TO MID COMP CYCLE	00002510
C		00002520
	THY=0.5*THY	00002530
	THZ=0.5*THZ	00002540
	CALL AUP(THY,THZ,M2D2O2)	00002550
C	WRITE (OFIL,551) M2D2O2	00002560
	CALL MM(A,M2D2O2,XA)	00002570
C	WRITE (OFIL,551) XA	00002580
C		00002590
C		00002600
C	VERTICAL DAMPING CALCULATIONS INCLUDING EXTRAPOLATION	00002610
C		00002620
	H = H + (HV(1) + CVD1*DELH)*DT	00002630
	XH = H + 0.5*V(1)*DT	00002640
	HB=ALTO	00002650
	XHB = 1.5*HB - 0.5*OHB	00002660
	DELH = XHB - XH	00002670
	OHB = HB	00002680
	VDMP = VDMP + CVD3*DELH*DT	00002690
	NAVH=H	00002700
C		00002710
C	EXTRACT LAT, LON, ALF, EXTRAPOLATE ALF	00002720
C		00002730
	CGDL=DSQRT(A(1,1)*A(1,1)+A(2,1)*A(2,1))	00002740
	NAVLAT=DATAN2(A(3,1),CGDL)	00002750
	OALF = ALF	00002760
	IF(CGDL .EQ. 0.) GOTO 115	00002770
	NAVLOH=DATAN2(A(2,1),A(1,1))	00002780
	ALF=DATAN2(A(3,2),A(3,3))	00002790
	DALF = 0.5*(ALF-OALF)	00002800
115	CALF=DCOS(ALF)	00002810
	SALF=DSIN(ALF)	00002820
	XSALF = SALF+ DALF*CALF	00002830
	XCALF = CALF- DALF*SALF	00002840
C		00002850
C	TRANSFORM VELOCITY TO ENU	00002860
C		00002870
	NAV(1)=CALF*V(2)-SALF*V(3)	00002880
	NAV(2)= SALF*V(2)+CALF*V(3)	00002890
	NAV(3)=V(1)	00002900
C		00002910
C	COMPUTE GRAVITY GR, EXTRAPOLATED TO MID COMP CYCLE	00002920
C		00002930
	CALL GRAV(XA,XH,XSSLAT,GR)	00002940
C	WRITE (OFIL,551) GR	00002950
	CALL ANGVEL(XV,XCALF,XSALF,XSSLAT,XH,RHO)	00002960
C	WRITE (OFIL,551) RHO	00002970
C		00002980
C	COMPUTE COMP TO INERTIAL RATES X DT AND CORIOLIS CORRECTIONS	00002990
C		00003000

	CALL TORCOR(XA,RHO,XV,DT,THG,WXV)	00003010
C	WRITE (OFILE,551) THG	00003020
C	WRITE (OFILE,551) WXV	00003030
C		00003040
C	ZERO DVI	00003050
C		00003060
	DO 140 J4=1,3	00003070
	DVI(J4)=0.	00003080
140	CONTINUE	00003090
C		00003100
C	COMPUTE BODY TO COMP FRAME TRANSFORMATION (ATTITUDE MATRIX)	00003110
C	AND EXTRACT ATTITUDE ANGLES	00003120
C		00003130
	CALL MTXM(AV,DCM,DTEM)	00003140
	DEN = SQRT(DTEM(2)**2+DTEM(3)**2)	00003150
	IF (DEN .EQ. 0.0) GO TO 950	00003160
	NAVP = ATAN2(DTEM(1),DEN)	00003170
	NAVR = ATAN2(DTEM(2),DTEM(3))	00003180
	NAVHD = ATAN2(DTEM(4),DTEM(7)) - ALF	00003190
	GO TO 940	00003200
950	WRITE (OFILE,1250)	00003210
	TLLN=T+DT	00003220
	RETURN	00003230
940	CONTINUE	00003240
C		00003250
C	OUTPUT AND PRINT CONTROL	00003260
C		00003270
	IF (PRNTDT.GT.0) GO TO 960	00003280
	IF (MODPDT.EQ.0) GO TO 999	00003290
C		00003300
	IF (T.LT.TPRMOD-.0005) GO TO 999	00003310
	TPRMOD=TPRMOD+MODPDT	00003320
	GO TO 970	00003330
C		00003340
960	CONTINUE	00003350
	IF (T.LT.TPRNT-.0005) GO TO 999	00003360
	TPRNT=TPRNT+PRNTDT	00003370
C		00003380
970	CONTINUE	00003390
	IF(PRNTSW.LT.1.) GOTO 999	00003400
C		00003410
	ITEMP1=NAVLAT*RDODG	00003420
	ITEMP2=NAVLOD*RDODG	00003430
	ITEMP3=NAVP * RDODG	00003440
	ITEMP4=NAVR * RDODG	00003450
	ITEMP5=NAVHD *RDODG	00003460
C		00003470
C		00003480
	WRITE (OFILE,1200) NAVH,NAVV,ITEMP1,ITEMP2,ITEMP3,ITEMP4,ITEMP5	00003490
	WRITE (OFILE,1225) A	00003500
	WRITE (OFILE,1235)DTEM	00003510
C		00003520
C		00003530
999	CONTINUE	00003540
	TLLN=T+DT	00003550
	RETURN	00003560
C		00003570
C	NAVIGATION MODULE INITIALIZATION	00003580
C		00003590
500	CONTINUE	00003600

C		00003610
C		00003620
	REWIND IFILE	00003630
	REWIND PFILE	00003640
501	READ (IFILE,1000) IX,DATA(IX)	00003650
	IF (EOF(IFILE)) 502,501	00003660
502	CONTINUE	00003670
503	READ (PFILE,1000) IX,PDATA(IX)	00003680
	IF (EOF(PFILE)) 510,503	00003690
510	CONTINUE	00003700
	REWIND IFILE	00003710
	REWIND PFILE	00003720
C		00003730
	OFIL=XFILE	00003740
C		00003750
C	INITIALIZATION FOR VERTICAL DAMPING	00003760
C		00003770
	H=IH+ALTERR	00003780
	HB= ALTO	00003790
	CHB= ALTO	00003800
	DELH=HB-H	00003810
	VDMF=CVD3*DELH*DT	00003820
	NUP=0	00003830
	NORTH=4	00003840
C	WRITE (OFIL,551) DELH,VDMF	00003850
C		00003860
C	COMPUTE INITIAL UEN-WA TO EARTH-FIXED TRANSFORM, A	00003870
C		00003880
	INAVLA=ILAT+(LATERR)/RDTODG	00003890
	INAVLO=ILON+(LONERR)/RDTODG	00003900
	SINLT=SIN(INAVLA)	00003910
	SSLAT=SINLT*SINLT	00003920
	SINLON=SIN(INAVLO)	00003930
	COSLON=COS(INAVLO)	00003940
	COSLT=COS(INAVLA)	00003950
	ALF=WONDER	00003960
	CALF=COS(WONDER)	00003970
	SALF=SIN(WONDER)	00003980
	A(1,1)=COSLT*COSLON	00003990
	A(2,1)=COSLT*SINLON	00004000
	A(3,1)=SINLT	00004010
	A(1,2)=-CALF*SINLON-SALF*SINLT*COSLON	00004020
	A(2,2)=COSLON*CALF-SALF*SINLT*SINLON	00004030
	A(3,2)= SALF*COSLT	00004040
	A(1,3)= SALF*SINLON-CALF*SINLT*COSLON	00004050
	A(2,3)=-SALF*COSLON-CALF*SINLT*SINLON	00004060
	A(3,3)=COSLT*CALF	00004070
C		00004080
	WRITE (OFIL,1225) A	00004090
C		00004100
C		00004110
C	COMPUTE INITIAL EARTH-RELATIVE VELOCITY IN UEN-WA FRAME	00004120
C		00004130
	DO 600 I=1,3	00004140
	NAVV(I)=IV(I)+VERR(I)	00004150
600	CONTINUE	00004160
	V(1)=NAVV(3)	00004170
	V(2)=CALF*NAVV(1) + SALF*NAVV(2)	00004180
	V(3)=-SALF*NAVV(1) + CALF*NAVV(2)	00004190
	XH=H+0.5*V(1)*DT	00004200

C		00004210
C	COMPUTE INITIAL ANGULAR VELOCITY OF UEN-WA FRAME WRT EARTH-FIXED	00004220
C	FRAME IN UEN-WA FRAME	00004230
	RHO(1)=0.0	00004240
	CALL ANGVEL(V,CALF,SALF,SSLAT,H,RHO)	00004250
C	WRITE (OFIL,551) RHO	00004260
C		00004270
C	COMPUTE COMP. TO INERTIAL FRAME RATE X DT AND CORIOLIS CORRECTIONS	00004280
C		00004290
	CALL TORCOR(A,RHO,V,DT,THG,WXV)	00004300
C	WRITE (OFIL,551) THG	00004310
C	WRITE (OFIL,551) WXV	00004320
C		00004330
C	EXTRAPOLATE A TO MID COMP CYCLE, XA	00004340
C		00004350
	THY = 0.5*RHO(2)*DT	00004360
	THZ = 0.5*RHO(3)*DT	00004370
	CALL AUP(THY,THZ,M2DT)	00004380
C	WRITE (OFIL,551) M2DT	00004390
	CALL MM(A,M2DT,XA)	00004400
C	WRITE (OFIL,551) XA	00004410
C		00004420
C	EXTRAPOLATE ALF TO MID COMP CYCLE	00004430
C		00004440
	DALF=(0.5*V(2)*SINLT*DT)/(RE*COSLT)	00004450
	XSALF=SALF + DALF*CALF	00004460
	XCALF=CALF - DALF*SALF	00004470
C		00004480
C	COMPUTE GRAVITY AT MID COMP CYCLE	00004490
C		00004500
	CALL GRAV(XA,XH,XSSLAT,GR)	00004510
C	WRITE (OFIL,551) GR	00004520
C		00004530
C	COMPUTE WXV AT MID COMP CYCLE	00004540
C		00004550
	CALL TORCOR(XA,RHO,V,DT,THG,WXV)	00004560
C		00004570
C	ZERO DVI	00004580
C		00004590
	DO 640 J4=1,3	00004600
	DVI(J4)=0.0	00004610
640	CONTINUE	00004620
C		00004630
C	INITIALIZATION OUTPUT AND PRINT CONTROL	00004640
C		00004650
	WRITE (OFIL,1010) DT,PRNTSW,OUTSW,OFIL,SPARE1,PRNTDT	00004660
	X,CVD1,CVD2,CVD3	00004670
C		00004680
	ITEMP1=ILAT*RDYODG	00004690
	ITEMP2=ILON*RDYODG	00004700
	WRITE(OFIL,1012) H,ALTERR,NAVV,VERR,ITEMP1,LATERR,ITEMP2,LONERR	00004710
C		00004720
C		00004730
	INITSW=1	00004740
	TLLN=T+DT	00004750
	RETURN	00004760
C		00004770
	551 FORMAT (3(/3X,3F20.16),/)	00004780
	1000 FORMAT (15,F20.10)	00004790
	1010 FORMAT(30H NAVIGATION INITIALIZATION ,	00004800

X/3X,8H DT	,3X,G16.8,3X,4H SEC,	00004810
X/3X,8H PRNTSW	,3X,G16.8,	00004820
X/3X,8H OUTSW	,3X,G16.8,	00004830
X/3X,8H OFILE	,3X,I15,	00004840
X/3X,8H SPARE	,3X,G16.8,	00004850
X/3X,8H PRNTDT	,3X,G16.8,	00004860
X/3X,12H CVD1(SEC-1)	,3X,G16.8,	00004870
X/3X,12H CVD2(SEC-2)	,3X,G16.8,	00004880
X/3X,12H CVD3(SEC-3)	,3X,G16.8,//)	00004890
1012 FORMAT(3X,30H INITIAL VEHICLE POSITION	,//	00004900
X	6X,11H H(FT) ,G16.8,/,	00004910
X	6X,11H ALTERR(FT),G16.8,/,	00004920
X	6X,11H V(FT/SEC) ,3G16.8,/,	00004930
X	6X,14H VERR(FT/SEC) ,3G16.8,/,	00004940
X	6X,11H LAT(DEGS) ,G16.8,/,	00004950
X	6X,11H LATERR(DEGS) ,G16.8,/,	00004960
X	6X,11H LON(DEGS) ,G16.8,/,	00004970
X	6X,13H LONERR(DEGS),G16.8,//)	00004980
1200 FORMAT(6X,25H ** LLN ** H(FT)	,G16.8,/,	00004990
X	6X,25H V(FT/SEC) ,3G16.8,/,	00005000
X	6X,25H LAT(DEGS) ,G16.8,/,	00005010
X	6X,25H LON(DEGS) ,G16.8,/,	00005020
X	6X,25H PITCH(DEGS) ,G16.8,/,	00005030
X	6X,25H ROLL(DEGS) ,G16.8,/,	00005040
X	6X,25H HEADING(DEGS) ,G16.8,/,	00005050
1225 FORMAT(6X,25H A TRANSPOSE	,3(/3X,3G16.8),/,)	00005060
1235 FORMAT(6X,25H DTEM	,3(/3X,3G16.8),/,)	00005070
1250 FORMAT(55H PITCH IS + OR -90DEGREES, PREVIOUS VALUES ARE OUTPUT		00005080
1300 FORMAT(10X,3F20.10,/,)		00005090
END		00005100

	SUBROUTINE ANGVEL (AVEL,ACALF,ASALF,ASSL,ALT,	00000010
	1 RHO)	00000020
C		00000030
C	COMPUTES LEVEL COMPONENTS OF ANGULAR VELOCITY OF VEHICLE	00000040
C	DUE TO ITS MOTION WRT EARTH	00000050
C		00000060
	REAL AVEL(3),ACALF,ASALF,ASSL,ALT,VE,VN,WE,WN,RM,RP	00000070
	REAL TMP1,RHO(3),ESQ,RESQ,RE,TMP2	00000080
C		00000090
	DATA ESQ/6.694317778E-3/	00000100
	DATA RESQ/2.078555712E7/	00000110
	DATA RE/2.0925640E7/	00000120
C		00000130
	VE = ACALF*AVEL(2) - ASALF*AVEL(3)	00000140
	VN = ASALF*AVEL(2) + ACALF*AVEL(3)	00000150
C		00000160
	TMP1=1.0 / (1.0 - ESQ*ASSL)	00000170
	TMP2=SQRT(TMP1)	00000180
C		00000190
	RP = ALT + RE*TMP2	00000200
	RM = ALT + RESQ*TMP1*TMP2	00000210
C		00000220
	WE = -VN/RM	00000230
	WN = VE/RP	00000240
C		00000250
	RHO(1) = 0.0	00000260
	RHO(2) = ACALF*WE + ASALF*WN	00000270
	RHO(3) = -ASALF*WE + ACALF*WN	00000280
	RETURN	00000290
	END	00000300

SUBROUTINE TORCOR(A,RHO,V,DT,	00000010
1 THG,WXV)	00000020
C	00000030
C COMPUTES ANGULAR VELOCITIES TIMES DT AND CORIOLIS CORRECTIONS	00000040
C	00000050
C REAL A(3,3),RHO(3),THG(3),WXV(3),THE(3),DT,WE,V(3),EAR(3)	00000060
C	00000070
C DATA WE/7.292115147E-5/	00000080
C	00000090
C DO 1 J=1,3	00000100
C	00000110
C CALCULATE EARTH RATE X DT(IN COMP.)	00000120
C	00000130
C EAR(J) = A(3,J)*WE*DT	00000140
C	00000150
C CALCULATE COMPUTATIONAL FRAME WRT INERTIAL FRAME RATE X DT(FOR QDICI)	00000160
C	00000170
C THG(J) = EAR(J) + RHO(J)*DT	00000180
C	00000190
C CALCULATE RATE X DT (FOR WXV)	00000200
C	00000210
C THE(J) = THG(J) + EAR(J)	00000220
C 1 CONTINUE	00000230
C	00000240
C CALCULATE CORIOLIS CORRECTIONS (FOR VELOCITY UPDATE)	00000250
C	00000260
C WXV(1) = THE(3)*V(2) - THE(2)*V(3)	00000270
C WXV(2) = THE(1)*V(3) - THE(3)*V(1)	00000280
C WXV(3) = THE(2)*V(1) - THE(1)*V(2)	00000290
C RETURN	00000300
C END	00000310

	SUBROUTINE AUP(DY,DZ,	00000010
	1 MUP)	00000020
C		00000030
C	FORM SECOND ORDER UPDATE MATRIX FOR LVNA COMP.	00000040
C	TO EARTH FIXED DCM	00000050
C		00000060
	REAL DY	00000070
	REAL DZ	00000080
	REAL MUP(3,3)	00000090
C		00000100
	MUP(1,1) = 1.0 - 0.5*(DY*DY+DZ*DZ)	00000110
	MUP(2,1) = DZ	00000120
	MUP(1,2) = -DZ	00000130
	MUP(1,3) = DY	00000140
	MUP(3,1) = -DY	00000150
	MUP(2,2) = 1.0 - 0.5* DZ*DZ	00000160
	MUP(3,2) = 0.5*DY*DZ	00000170
	MUP(2,3) = MUP(3,2)	00000180
	MUP(3,3) = 1.0 - 0.5*DY*DY	00000190
	RETURN	00000200
	END	00000210

	SUBROUTINE GRAV (AM,ALT,	00000010
	1 SSL,G)	00000020
C		00000030
C	COMPUTES 3 COMPONENTS OF GRAVITY IN COMPUTATIONAL FRAME	00000040
C	USING WGS-72 ELLIPSOIDAL EARTH MODEL	00000050
C		00000060
	REAL AM(3,3),G(3),SSL,ALT,COEF	00000070
C		00000080
	SSL = AM(3,1)*AM(3,1)	00000090
	COEF = 1.63E-8*ALT*AM(3,1)	00000100
C		00000110
	G(1) = -(32.0877057 + .16939081*SSL	00000120
1	+ 7.5281E-4*SSL*SSL)*(1.0-(9.6227E-6	00000130
2	- 6.4089E-10*SSL*ALT+6.8512E-15*ALT*ALT)	00000140
C		00000150
	G(2) = COEF*AM(3,2)	00000160
C		00000170
	G(3) = COEF*AM(3,3)	00000180
C		00000190
	RETURN	00000200
	END	00000210

```

      SUBROUTINE MM(A,B,C)
C
C  MATRIX MULTIPLY SUBROUTINE
C  (3X3) X (3X3) = (3X3) ; A*B=C
C
      REAL A(3,3),B(3,3),C(3,3)
      DO 1 I=1,3
      DO 1 J=1,3
      C(I,J) =0.0
      DO 1 K=1,3
      C(I,J) = C(I,J) + A(I,K)*B(K,J)
1  CONTINUE
      RETURN
      END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140

```


C		00000010
C	03/10/78 DATE OF CURRENT MODULE	00000020
C		00000030
C	THE EVALUATION MODULE PRINTS TABLES OF NAVIGATION ERRORS AND	00000040
C	TRAJECTORY PARAMETERS.	00000050
C		00000060
	SUEROUTINE EVL(T,IENDF,LAT,LON,ALT,VEL,DVT,PITCH,ROLL,YAW,	00000070
	X WONDER,NAVLAT,NAVLOX,NAVY,NAVH,NAVZ,NAVR,NAVHD)	00000080
C		00000090
	REAL ALT	00000100
	REAL DATA(5)	00000110
	REAL DT	00000120
	REAL DVT(3)	00000130
	REAL EALT(50)	00000140
	REAL ELAT(50)	00000150
	REAL ELON(50)	00000160
	REAL EVELX(50)	00000170
	REAL EVELY(50)	00000180
	REAL EVELZ(50)	00000190
	REAL LAT	00000200
	REAL LON	00000210
	REAL NAVLAT	00000220
	REAL NAVLOX	00000230
	REAL NAVY	00000240
	REAL NAVZ(3)	00000250
	REAL PRMTOT	00000260
	REAL ROTODG	00000270
	REAL T	00000280
	REAL TEVL	00000290
	REAL VEL(3)	00000300
	REAL XALT(50)	00000310
	REAL XDVTX(50)	00000320
	REAL XDVTY(50)	00000330
	REAL XDVTZ(50)	00000340
	REAL XLAT(50)	00000350
	REAL XLON(50)	00000360
	REAL XVELX(50)	00000370
	REAL XVELY(50)	00000380
	REAL XVELZ(50)	00000390
	REAL XT(50)	00000400
	REAL XHEAD(50)	00000410
	REAL XPITCH(50)	00000420
	REAL XROLL(50)	00000430
	REAL EHEAD(50)	00000440
	REAL EPITCH(50)	00000450
	REAL EROLL(50)	00000460
C		00000470
	REAL NAVP	00000480
	REAL NAVR	00000490
	REAL NAVHD	00000500
	REAL PITCH	00000510
	REAL ROLL	00000520
	REAL YAW	00000530
	REAL WONDER	00000540
	REAL PDATA(20)	00000550
	REAL WE	00000560
	REAL RE	00000570
	REAL G	00000580
C		00000590
	INTEGER IENDF	00000600

	INTEGER INITSW	00000610
	INTEGER IFILE	00000620
	INTEGER PFILE	00000630
	INTEGER N	00000640
	INTEGER OFILE	00000650
	INTEGER OUTSW	00000660
	INTEGER PPFILE	00000670
C		00000680
	EQUIVALENCE (DATA(1), DT)	00000690
	EQUIVALENCE (DATA(2), PRNTSW)	00000700
	EQUIVALENCE (DATA(3), OUTSW)	00000710
	EQUIVALENCE (DATA(4), XFILE)	00000720
	EQUIVALENCE (DATA(5), PRNTDT)	00000730
C		00000740
	EQUIVALENCE (PDATA(1), WE)	00000750
	EQUIVALENCE (PDATA(2), RE)	00000760
	EQUIVALENCE (PDATA(3), G)	00000770
C		00000780
	DATA RDTODG/57.29577951/	00000790
	DATA INITSW/0/	00000800
	DATA IFILE/90/	00000810
	DATA N/C/	00000820
	DATA PFILE/7/	00000830
	DATA PPFILE/12/	00000840
C		00000850
	IF (IENDF.GT.0) GO TO 800	00000860
	IF (INITSW.EQ.0) GO TO 500	00000870
	IF (T.LT.TEVL*.0005) RETURN	00000880
C		00000890
	N=N+1	00000900
C		00000910
	XT(N)=T	00000920
	XLAT(N)=LAT*RDTODG	00000930
	XLON(N)=LON*RDTODG	00000940
	XALT(N)=ALT	00000950
	XVELX(N)=VEL(1)	00000960
	XVELY(N)=VEL(2)	00000970
	XVELZ(N)=VEL(3)	00000980
	XOVTX(N)=OVT(1)	00000990
	XOVTY(N)=OVT(2)	00001000
	XOVTZ(N)=OVT(3)	00001010
	XHEAD(N)=(YAW-WONDER)*RDTODG	00001020
	XPITCH(N)=PITCH*RDTODG	00001030
	XROLL(N)=ROLL*RDTODG	00001040
C		00001050
	ELAT(N)=(LAT-NAVLAT)*RE	00001060
	ELON(N)=(LON-NAVLON)*RE*COS(LAT)	00001070
	EALT(N)=ALT-NAVH	00001080
	EVELX(N)=VEL(1)-NAVV(1)	00001090
	EVELY(N)=VEL(2)-NAVV(2)	00001100
	EVELZ(N)=VEL(3)-NAVV(3)	00001110
	EHEAD(N)=(YAW-WONDER-NAVH)*3600*RDTODG	00001120
	EPITCH(N)=(PITCH-NAVP)*3600*RDTODG	00001130
	EROLL(N)=(ROLL-NAVR)*3600*RDTODG	00001140
C		00001150
C	WRITE PLOT FILE (PPFILE)	00001160
	WRITE (PPFILE) XT(N),XLAT(N),XLON(N),XALT(N),XVELX(N),XVELY(N),	00001170
	X XVELZ(N),XHEAD(N),XPITCH(N),XROLL(N),ELAT(N),ELON(N),EALT(N),	00001180
	X EVELX(N),EVELY(N),EVELZ(N),EHEAD(N),EPITCH(N),EROLL(N)	00001190
C		00001200

IF (N.LT.49.5) GO TO 900	0001210
C	0001220
800 CONTINUE	0001230
C	0001240
C OUTPUT AND PRINT CONTROL	0001250
C (A TABLE IS PRINTED EVERY 50 CYCLES)	0001260
C LAST PASS COMES HERE TO PRINT ANY REMAINING DATA	0001270
C	0001280
IF (PRNTSW.LT.1.) GO TO 999	0001290
C	0001300
WRITE (OFILE,1200)	0001310
WRITE (OFILE,1300)	0001320
WRITE (OFILE,1320)	0001330
C	0001340
DO 810 I=1,N	0001350
WRITE (OFILE,1400)XT(I),XLAT(I),XLON(I),XALT(I),XVELX(I),XVELY(I),	0001360
X XVELZ(I),XHEAD(I),XPITCH(I),XROLL(I)	0001370
810 CONTINUE	0001380
C	0001390
WRITE (OFILE,1250)	0001400
WRITE (OFILE,1300)	0001410
WRITE (OFILE,1330)	0001420
C	0001430
DO 820 I=1,N	0001440
WRITE (OFILE,1400)XT(I),ELAT(I),ELON(I),EALT(I),EVELX(I),EVELY(I),	0001450
X EVELZ(I),EHEAD(I),EPITCH(I),EROLL(I)	0001460
820 CONTINUE	0001470
C	0001480
N=0	0001490
C	0001500
C	0001510
900 CONTINUE	0001520
C	0001530
999 CONTINUE	0001540
TEVL=V*DT	0001550
RETURN	0001560
C	0001570
C	0001580
C EVALUATION MODULE INITIALIZATION	0001590
C	0001600
500 CONTINUE	0001610
REWIND IFILE	0001620
REWIND PFILE	0001630
501 READ (IFILE,1000) IX,DATA(IX)	0001640
IF (EOF(IFILE)) 502,501	0001650
502 CONTINUE	0001660
REWIND IFILE	0001670
503 READ (PFILE,1000) IX,PODATA(IX)	0001680
IF (EOF(PFILE)) 510,503	0001690
510 CONTINUE	0001700
REWIND PFILE	0001710
C	0001720
C	0001730
C INITIALIZATION OUTPUT AND PRINT CONTROL	0001740
C	0001750
OFILE=XFILE	0001760
WRITE (OFILE,1100) DT,PRNTSW,OUTSW,OFILE,PRNTDT	0001770
C	0001780
C	0001790
INITSW=1	0001800

TEVL=T+DT	00001810
RETURN	00001820
C	00001830
1000 FORMAT(I5,F20.10)	00001840
C	00001850
1100 FORMAT(30H EVALUATION INITIALIZATION ,	00001860
X/3X,8H DT ,3X,G16.6,3X,4H SEC,	00001870
X/3X,8H PRNTSW ,3X,G16.6,	00001880
X/3X,8H OUTSW ,3X,G16.6,	00001890
X/3X,8H OFILE ,I15,	00001900
X/3X,8H PRNTOT ,3X,G16.6,/))	00001910
C	00001920
1200 FORMAT(1H1,36X,11H TRAJECTORY)	00001930
C	00001940
1250 FORMAT(1H1,36X,18H NAVIGATION ERRORS)	00001950
C	00001960
1300 FORMAT(2X,5H TIME,3X,8H LAT,5X,8H LON,5X,8H ALT,5X,	00001970
X 9H VX(EAST),4X,10H VY(NORTH),3X,8H VZ(UP),5X,8H HEAD,5X,	00001980
X 8H PITCH,5X,8H ROLL)	00001990
C	00002000
1320 FORMAT(8H (SEC),2X,8H (DEG),5X,8H (DEG),5X,8H (FT),5X,	00002010
X 8H (FPS),5X,8H (FPS),5X,8H (FPS),5X,8H (DEG),5X,	00002020
X 8H (DEG),5X,8H (DEG),/)	00002030
C	00002040
1330 FORMAT(8H (SEC),2X,8H (FT),5X,8H (FT),5X,8H (FT),5X,	00002050
X 8H (FPS),5X,8H (FPS),5X,8H (FPS),5X,8H (SEC),5X,	00002060
X 8H (SEC),5X,8H (SEC),/)	00002070
C	00002080
1400 FORMAT(F9.3,1X,9(G12.5,1X))	00002090
C	00002100
1500 FORMAT(10X,I15)	00002110
END	00002120

C	FUNCTION GAUSS	00000010
C		00000020
C	THIS FUNCTION CALCULATES A GAUSSIAN RANDOM NUMBER	00000030
C	IT IS MACHINE INDEPENDENT AND WILL WORK PROPERLY IF THE SINGLE	00000040
C	PRECISION WORD OF THE MACHINE IS MORE THAN 20 BITS LONG	00000050
C		00000060
C		00000070
	FUNCTION GAUSS (MEAN ,STD)	00000080
	REAL MEAN,STD	00000090
	DATA I/O/	00000100
	IF(I.EQ.1) GOTO 10	00000110
	IX = 3	00000120
	I = 1	00000130
10	CONTINUE	00000140
	FA = 0.	00000150
	DO 40 J = 1,12	00000160
	IY = IX * 1029 + 110795	00000170
	IY = MOD(IY,524280)	00000180
	FY = IY	00000190
	FA = FY * 1.907349E-6 + FA	00000200
	IX = IY	00000210
40	CONTINUE	00000220
	GAUSS = MEAN + STD*(FA - 6.)	00000230
	RETURN	00000240
	END	00000250
C		00000260
C		00000270

```

SUBROUTINE MTXH(MA,MB,MC)
REAL*8 MA(9),MB(9),MC(9)
REAL*8 VIN(3),V(3)
DO 10 I=1,3
VIN(I)=MB(I)
VIN(2)=MB(I+3)
VIN(3)=MB(I+6)
CALL MTXV (MA,VIN,V)
MC(I) = V(1)
MC(I+3) = V(2)
MC(I+6) = V(3)
10 CONTINUE
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150

```

```

SUBROUTINE MTXV(M,V,MV)
REAL*8 M(9),V(3),MV(3)
MV(1) = M(1)*V(1) + M(4)*V(2) + M(7)*V(3)
MV(2) = M(2)*V(1) + M(5)*V(2) + M(8)*V(3)
MV(3) = M(3)*V(1) + M(6)*V(2) + M(9)*V(3)
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080

```

SUBROUTINE MXM(MA,MB,MAB)
REAL*8 MA(9),MB(9),MAB(9)
CALL MTXV (MB,MA(1),MAB(1))
CALL MTXV (MB,MA(4),MAB(4))
CALL MTXV (MB,MA(7),MAB(7))
RETURN
END

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080


```

SUBROUTINE MXV(M,V1,V2)
REAL*8 M(9),V1(3),V2(3)
V2(1) = M(1)*V1(1) + M(2)*V1(2) + M(3)*V1(3)
V2(2) = M(4)*V1(1) + M(5)*V1(2) + M(6)*V1(3)
V2(3) = M(7)*V1(1) + M(8)*V1(2) + M(9)*V1(3)
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080

```

R-1136

VOLUME I, II, III, IV

DISTRIBUTION LIST

A. Ciccolo
G. Coate
R. Crisp
K. Daly
M. Dare
W. Delaney
W. Denhard
J. DiSorbo
J. Feldman
K. Fertig
J. Fish
J. Gilmore
J. Goode
J. Harper
R. Harris
J. Hursh
P. Kampion
A. Kosol
W. Koenigsberg
R. Leger
J. Lombardo
D. Millard
B. McCoy
P. Motyka
R. Nurse (19)

J. Prohaska
T. Reed
D. Riegsecker
J.P. Ryan
G. Schmidt
J. Sciegieny
L. Schnee
R. Setterlund
T. Thorvaldsen
L.W. Torrey
K. Vincent
P. Volante
TIC (5)

Air Force Avionics Laboratory (30)
Attn: RWA-3 D. Kaiser
Wright-Patterson Air Force Base
Dayton, Ohio 45433

Air Force Avionics Laboratory (5)
Attn: N. Banke RWA-2
Wright-Patterson Air Force Base
Dayton, Ohio 45433